F28E12x Real-Time Microcontrollers

Technical Reference Manual

Literature Number: SPRUJD3A
JULY 2025 – REVISED OCTOBER 2025

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

About This Manual

This Technical Reference Manual (TRM) details the integration, the environment, the functional description, and the programming models for each peripheral and subsystem in the device.

The TRM should not be considered a substitute for the data manual, rather a companion guide that should be used alongside the device-specific data manual to understand the details to program the device. The primary purpose of the TRM is to abstract the programming details of the device from the data manual. This allows the data manual to outline the high-level features of the device without unnecessary information about register descriptions or programming models.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers can be shown with the suffix h or the prefix 0x. For example, the following number is 40 hexadecimal (decimal 64): 40h or 0x40.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties with default reset value below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure can have one of multiple meanings:
    - Not implemented on the device
    - Reserved for future device expansion
    - Reserved for TI testing
    - Reserved configurations of the device that are not supported
  - Writing nondefault values to the Reserved bits could cause unexpected behavior and should be avoided.

Glossary

TI Glossary	This glossary lists and explains terms, acronyms, and definitions.

Related Documentation From Texas Instruments

For a complete listing of related documentation and development-support tools for these devices, visit the Texas Instruments website at www.ti.com.

Additionally, the *TMS320C28x DSP CPU and Instruction Set Reference Guide* and the *TMS320C28x Floating Point Unit and Instruction Set Reference Guide* must be used in conjunction with this TRM.

Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

Trademarks

TI E2E™, C2000™, Code Composer Studio™, and Texas Instruments™ are trademarks of Texas Instruments. All trademarks are the property of their respective owners.

Chapter 1

C2000™ Microcontrollers Software Support

This chapter discusses the C2000Ware for the C2000™ microcontrollers. The C2000Ware can be downloaded from: www.ti.com/tool/C2000WARE.

1.1 Introduction

C2000Ware for the C2000™ microcontrollers is a cohesive set of development software and documentation designed to minimize software development time. From device-specific drivers and libraries to device peripheral examples, C2000Ware provides a solid foundation to begin development and evaluation of your product.

C2000Ware can be downloaded from: www.ti.com/tool/C2000WARE

1.2 C2000Ware Structure

The C2000Ware software package is organized into the following directory structure as shown in Table 1-1.

Table 1-1. C2000Ware Root Directories

Directory Name	Description
boards	Contains the hardware design schematics, BOM, Gerber files, and documentation for C2000 controlCARDS.
device_support	Contains all device-specific support files, bit field headers and device development user's guides.
docs	Contains the C2000Ware package user's guides and the HTML index page of all package documentation.
driverlib	Contains the device-specific driver library and driver-based peripheral examples.
libraries	Contains the device-specific and core libraries.

1.3 Documentation

Within C2000Ware, there is an extensive amount of development documentation ranging from board design documentation, to library user's guides, to driver API documentation. The "boards" directory contains all the hardware design, BOM, Gerber files, and more for controlCARDs. To assist with locating the necessary documentation, an HTML page is provided that contains a full list of all the documents in the C2000Ware package. Locate this page in the "docs" directory.

1.4 Devices

C2000Ware contains the necessary software and documentation to jumpstart development for C2000™ microcontrollers. Each device includes device-specific common source files, peripheral example projects, bit field headers, and if available, a device peripheral driver library. Additionally, documentation is provided for each device on how to set up a CCS project, as well as give an overview of all the included example projects and assist with troubleshooting. For devices with a driver library, documentation is also included that details all the peripheral APIs available.

To learn more about C2000™ microcontrollers, visit: www.ti.com/c2000.

1.5 Libraries

The libraries included in C2000Ware range from fixed-point and floating-point math libraries, to specialized DSP libraries, as well as calibration libraries. Each library includes documentation and examples, where applicable. Additionally, the Flash API files and boot ROM source code are located in the "libraries" directory.

1.6 Code Composer Studio™ Integrated Development Environment (IDE)

Code Composer Studio™ is an integrated development environment (IDE) that supports TI's microcontroller and embedded processors portfolio. The Code Composer Studio™ IDE comprises a suite of tools used to develop and debug embedded applications. The latest version of Code Composer Studio™ IDE can be obtained at: www.ti.com/ccstudio

All projects and examples in C2000Ware are built for and tested with the Code Composer Studio™ IDE. Although the Code Composer Studio™ IDE is not included with the C2000Ware installer, Code Composer Studio™ IDE is easily obtainable in a variety of versions.

1.7 SysConfig and PinMUX Tool

To help simplify configuration challenges and accelerate software development, Texas Instruments™ created SysConfig, an intuitive and comprehensive collection of graphical utilities for configuring pins, peripherals, subsystems, and other components. SysConfig helps you manage, expose, and resolve conflicts visually so that you have more time to create differentiated applications.

The tool's output includes C header and code files that can be used with C2000Ware examples or used to configure custom software.

The SysConfig tool automatically selects the pinmux settings that satisfy the entered requirements. The SysConfig tool is delivered integrated in the Code Composer Studio™ IDE, in the C2000Ware GPIO example, as a standalone installer, or can be used by way of the cloud tools portal at: dev.ti.com

Chapter 2

C28x Processor

This chapter contains a short description of the C28x processor and extended instruction sets.

Further information can be found in the following documents:

- *TMS320C28x CPU and Instruction Set Reference Guide*
- *TMS320C28x Extended Instruction Sets Technical Reference Manual*
- *Accelerators: Enhancing the Capabilities of the C2000 MCU Family Technical Brief*
- *TMS320C28x FPU Primer Application Report*

2.1 Introduction

The C28x CPU is a 32-bit fixed-point processor. This device draws from the best features of digital signal processing, reduced instruction set computing (RISC), microcontroller architectures, firmware, and tool sets.

For more information on CPU architecture and instruction set, see the *TMS320C28x CPU and Instruction Set Reference Guide*.

2.2 C28X Related Collateral

Foundational Materials

- C2000 C28x Migration from COFF to EABI

2.3 Features

The CPU features include a modified Harvard architecture and circular addressing. The RISC features are single-cycle instruction execution, register-to-register operations, and modified Harvard architecture. The microcontroller features include ease of use through an intuitive instruction set, byte packing and unpacking, and bit manipulation. The modified Harvard architecture of the CPU enables instruction and data fetches to be performed in parallel. The CPU can read instructions and data while the CPU writes data simultaneously to maintain the single-cycle instruction operation across the pipeline.

2.4 Floating-Point Unit (FPU)

The C28x plus floating-point (C28x+FPU) processor extends the capabilities of the C28x fixed-point CPU by adding registers and instructions to support IEEE single-precision floating point operations.

Devices with the C28x+FPU include the standard C28x register set plus an additional set of floating-point unit registers. The additional floating-point unit registers are the following:

- Eight floating-point result registers, RnH (where n = 0–7)
- Floating-point Status Register (STF)
- Repeat Block Register (RB)

All of the floating-point registers, except the repeat block register, are shadowed. This shadowing can be used in high-priority interrupts for fast context save and restore of the floating-point registers.

For more information, see the *TMS320C28x Extended Instruction Sets Technical Reference Manual*.

Chapter 3

System Control and Interrupts

The system-level functionality of this microcontroller configures the clocking, resets, and interrupts of the CPU and peripherals, as well as the operation of the on-chip memories, timers, and security features.

3.1 Introduction

System-level configuration is controlled by a group of submodules that are collectively referred to as the system control module. The system control module provides the following capabilities:

- System-level resets, including power-on and brownout resets
- Clock source selection and PLL configuration
- Missing clock detection
- Clock-gating low-power modes
- Peripheral interrupt handling
- Non-maskable interrupts for certain fault conditions
- Three 32-bit timers
- Windowed watchdog timer, which can generate an interrupt or a reset
- RAM initialization, write protection, and mastership control
- Flash memory ECC, wait state, and cache configuration
- Dual-zone code security module (DCSM)

3.1.1 SYSCTL Related Collateral

Foundational Materials

- *C2000 MCU JTAG Connectivity Debug Application Report*

Getting Started Materials

- C2000™ SysConfig: ClockTree tool (Video)
- Debugging JTAG
- XDS Target Connection Guide

Expert Materials

- *C2000 CPU Memory Built-In Self-Test Application Report*
- *Programming of External Nonvolatile Memory Using SDFlash for TMS320C28x Devices Application Report*
- *Software Phased-Locked Loop (PLL) Design Using C2000 Microcontrollers Application Report*

3.1.2 LOCK Protection on System Configuration Registers

Several system configuration registers are protected from spurious CPU writes by “LOCK” registers. Once these associated LOCK register bits are set, the respective locked registers can no longer be modified by software. See the register descriptions for details.

3.1.3 EALLOW Protection

Some registers in the system are protected from spurious CPU or DMA writes by the EALLOW protection mechanism. This uses the special CPU instructions EALLOW and EDIS to enable and disable access to protected registers. The current protection state is given by the EALLOW bit in the CPU ST1 register, as shown in Table 3-1.

Register protection is enabled by default at startup. While protected, all writes to protected registers by the CPU or DMA are ignored. Only CPU reads, DMA reads, JTAG reads, and JTAG writes are allowed. If protection is disabled by executing the EALLOW instruction, the CPU and DMA are allowed to write freely to protected registers. After modifying registers, the registers can once again be protected by executing the EDIS instruction to clear the EALLOW bit.

Writes to the clock configuration and peripheral clock enable registers can be disabled until the next reset by writing to special lock registers.

Table 3-1. Access to EALLOW-Protected Registers

EALLOW Bit	CPU Writes	CPU Reads	DMA Writes	DMA Reads	JTAG Writes	JTAG Reads
0	Ignored	Allowed	Ignored	Allowed	Allowed (1)	Allowed
1	Allowed	Allowed	Allowed	Allowed	Allowed	Allowed

(1) The EALLOW bit is overridden by way of the JTAG port, allowing full access of protected registers during debug from the Code Composer Studio™ interface.

3.2 Power Management

The TMS320F28E12x MCU 1.2V core is powered up internally and is derived from the 3.3V rail.

3.3 Device Identification and Configuration Registers

The device identification registers and configuration registers provide information on the part number, revision, qualification status, and feature availability of the device.

All of the device information is part of the DEV_CFG_REGS space. The identification registers are PARTIDL, PARTIDH, and REVID.

A 256-bit Unique ID (UID) is available in UID_REGS. The 256 bits are separated into these registers:

- UID_PSRAND0-4: 160 bits of pseudo-random data
- UID_UNIQUE: 64-bit unique data; the value in this register is unique across all devices in the same PARTIDH
- UID_CHECKSUM: 32-bit Fletcher checksum of UID_PSRAND0-4 and UID_UNIQUE and calculated as either little- or big-endian during factory testing

3.4 Resets

This section explains the types and effects of the different resets on this device.

3.4.1 Reset Sources

Table 3-2 summarizes the various reset signals and the effect on the device.

Table 3-2. Reset Signals

Reset Source	CPU Core Reset (C28x, FPU)	Peripherals Reset	JTAG / Debug Logic Reset	IOs	$\overline{\text{XRS}}$ Output
POR	Yes	Yes	Yes	Hi-Z	Yes
BOR	Yes	Yes	Yes	Hi-Z	Yes
$\overline{\text{XRS}}$ Pin	Yes	Yes	No	Hi-Z	-
$\overline{\text{WDRS}}$	Yes	Yes	No	Hi-Z	Yes
$\overline{\text{NMIWDRS}}$	Yes	Yes	No	Hi-Z	Yes
$\overline{\text{SYSRS}}$ (Debugger Reset)	Yes	Yes	No	Hi-Z	No
$\overline{\text{SCCRESET}}$	Yes	Yes	No	Hi-Z	No
SIMRESET. $\overline{\text{XRS}}$	Yes	Yes	No	Hi-Z	Yes
SIMRESET. $\overline{\text{CPU1RS}}$	Yes	Yes	No	Hi-Z	No

The resets can be divided into two groups:

- Chip-level resets ($\overline{\text{XRS}}$, POR, BOR, $\overline{\text{WDRS}}$, SIMRESET.$\overline{\text{XRS}}$ and $\overline{\text{NMIWDRS}}$), which reset all or almost all of the device.
- System resets ($\overline{\text{SYSRS}}$, SIMRESET.$\overline{\text{CPU1RS}}$ and $\overline{\text{SCCRESET}}$), which reset a large subset of the device but maintain some system-level configuration.

After a reset, the reset cause register (RESC) is updated with the reset cause. The bits in this register maintain the state across multiple resets. The bits can only be cleared by a power-on reset (POR) or by writing ones to the RESCCLR register. Some are cleared by the boot ROM as part of the start-up routines.

Many peripheral modules have individual resets accessible through the SOFTPRESx registers. For information about a module's reset state, refer to the chapter for that module.

After any reset, the CPU begins execution from address 0x3FFFC0 (the reset vector), which is in the boot ROM. After running the boot ROM code, the CPU typically branches to the start of the Flash memory at address 0x80000. For more information on controlling the boot process, see *ROM Code and Peripheral Booting* chapter.

Note

After a POR, the boot ROMs clear the M0/M1 and GSx RAMs to make sure that the RAMs contain valid parity.

3.4.2 External Reset ($\overline{XRS}$)

The external reset ($\overline{XRS}$) is the main chip-level reset for the device. The $\overline{XRS}$ resets the CPU, all peripherals and I/O pin configurations, and most of the system control registers. There is a dedicated open-drain pin for $\overline{XRS}$. This pin can be used to drive reset pins for other ICs in the application, and can be driven by an external source. The $\overline{XRS}$ is driven internally during watchdog, NMI, and power-on resets.

The XRSn bit in the RESC register is set whenever $\overline{XRS}$ is driven low for any reason. This bit is then cleared by the boot ROM.

3.4.3 Power-On Reset (POR)

The power-on reset (POR) circuit creates a clean reset throughout the device during power-up, suppressing glitches on the GPIOs. The $\overline{XRS}$ pin is held low for the duration of the POR. In most applications, $\overline{XRS}$ is held low long enough to reset other system ICs, but some applications can require a longer pulse. In these cases, the $\overline{XRS}$ pin can be driven low externally to provide the correct reset duration. A POR resets everything that $\overline{XRS}$ does, along with a few other registers – the reset cause register (RESC), the NMI shadow flag register (NMISHDFLG), and the X1 clock counter register (X1CNT). A POR also resets the debug logic used by the JTAG port.

After a POR, the POR and XRSn bits in RESC are set. These bits are then cleared by the boot ROM.

3.4.4 Brown-Out-Reset (BOR)

The brown-out-reset (BOR) is an internal supply voltage supervisor (SVS) circuit which monitors the VDDIO supply for glitches or supply interruptions. If the VDDIO supply voltage drops below operational voltage range, this circuit forces the XRSn pin low until the fault is removed and the supply voltage returns to the minimum operational voltage. A BOR resets everything in the same manner as a POR reset.

The BOR circuit is enabled by default and therefore is always active during power up or after any type of reset. To disable the BOR circuit, set the BORLVMONDIS bit in the VMONCTL register.

3.4.5 Watchdog Reset ($\overline{WDRS}$)

The device has a watchdog timer that can optionally trigger a reset, if the watchdog timer is not serviced by the CPU within a user-specified amount of time. This watchdog reset ($\overline{WDRS}$) produces an $\overline{XRS}$ that lasts for 512 SECOSC cycles.

After a watchdog reset, the WDRSn and XRSn bits in RESC are set.

3.4.6 NMI Watchdog Reset (NMIWDRS)

The device has a non-maskable interrupt (NMI) module that detects hardware errors in the system. The NMI module has a watchdog timer that triggers a reset if the CPU does not respond to an error within a user-specified amount of time. This NMI watchdog reset (NMIWDRS) produces an XRS that lasts for 512 SECOSC cycles.

After an NMI watchdog reset, the NMIWDRSn and XRSn bits in RESC are set.

3.4.7 Debugger Reset (SYSRS)

During development, resetting the CPU and the peripherals without disconnecting the debugger or disrupting the system-level configuration is sometimes necessary. To facilitate this, the CPU has a subsystem reset, which can be triggered by a debugger using the Code Composer Studio™ IDE. This reset (SYSRS) resets the CPU, the peripherals, many system control registers (including the clock gating and LPM configuration), and all I/O pin configurations.

The SYSRS does not reset the ICEPick debug module, the device capability registers, the clock source and PLL configurations, the missing clock detection state, the PIE vector fetch error handler address, the NMI flags, the analog trims, or anything reset only by a POR (see Section 3.4.3).

3.4.8 DCSM Safe Code Copy Reset (SCCRESET)

The device has a dual-zone code security module (DCSM) that blocks read access to certain areas of the Flash memory. To facilitate CRC checks, TI provides ROM functions to securely access those memory areas. To prevent security breaches, interrupts must be disabled before calling these functions. If a vector fetch occurs in a safe copy or CRC function, the DCSM triggers a reset. This security reset (SCCRESET) is similar to a SYSRS. However, the security reset also resets the debug logic to deny access to a potential attacker.

After a security reset, the SCCRESETn bit in RESC is set.

3.4.9 Simulate External Reset (SIMRESET.XRS)

The user can simulate an external reset (XRS) in software. This can be done by setting the XRSn bit to 1 in the SIMRESET register by software. This toggles the XRS pin; hence, resetting the full device (just like external reset).

After this reset, the SIMRESET_XRSn and XRSn bits in the RESC register are set. Software can read these bits to know the cause of the reset and clear the status by writing a 1 into the corresponding bits in the RESCCLR register.

3.4.10 Simulate CPU Reset (SIMRESET_CPU1RS)

The user can simulate a CPU reset (SYSRS) in software. This can be done by setting CPU1RSn bit to 1 in the SIMRESET register by CPU1 software. This toggles the CPU1.SYSRS signals; hence, resetting the CPU (just like the debugger reset).

After this reset, the SIMRESET_CPU1RSn bit in the RESC register is set. Software can read this bit to know the cause of the reset and clear the status by writing a 1 into the corresponding bit in the RESCCLR register.

3.5 Peripheral Interrupts

This section explains the peripheral interrupt handling on the device. Non-maskable interrupts are covered in Section 3.6. Software interrupts and emulation interrupts are not covered in this document. For information on those, see the *TMS320C28x CPU and Instruction Set Reference Guide*.

3.5.1 Interrupt Concepts

An interrupt is a signal that causes the CPU to pause the current execution and branch to a different piece of code known as an interrupt service routine (ISR). This is a useful mechanism for handling peripheral events, and involves less CPU overhead or program complexity than register polling. However, because interrupts are asynchronous to the program flow, care must be taken to avoid conflicts over resources that are accessed both in interrupts and in the main program code.

Interrupts propagate to the CPU through a series of flag and enable registers. The flag registers store the interrupt until the interrupt is processed. The enable registers block the propagation of the interrupt. When an interrupt signal reaches the CPU, the CPU fetches the appropriate ISR address from a list called the vector table.

3.5.2 Interrupt Architecture

The C28x CPU has 14 peripheral interrupt lines. Two of them (INT13 and INT14) connected directly to CPU timers 1 and 2, respectively. The remaining 12 interrupt lines are connected to peripheral interrupt signals through the enhanced Peripheral Interrupt Expansion module (ePIE, or PIE as a shortened version). The PIE multiplexes up to four peripheral interrupts into each CPU interrupt line. The PIE also expands the vector table to allow each interrupt to have an ISR. This allows the CPU to support a large number of peripherals.

An interrupt path is divided into three stages – the peripheral, the PIE, and the CPU. Each stage has enable and flag registers. This system allows the CPU to handle one interrupt while others are pending, implement and prioritize nested interrupts in software, and disable interrupts during certain critical tasks.

Figure 3-1 shows the interrupt architecture for this device.

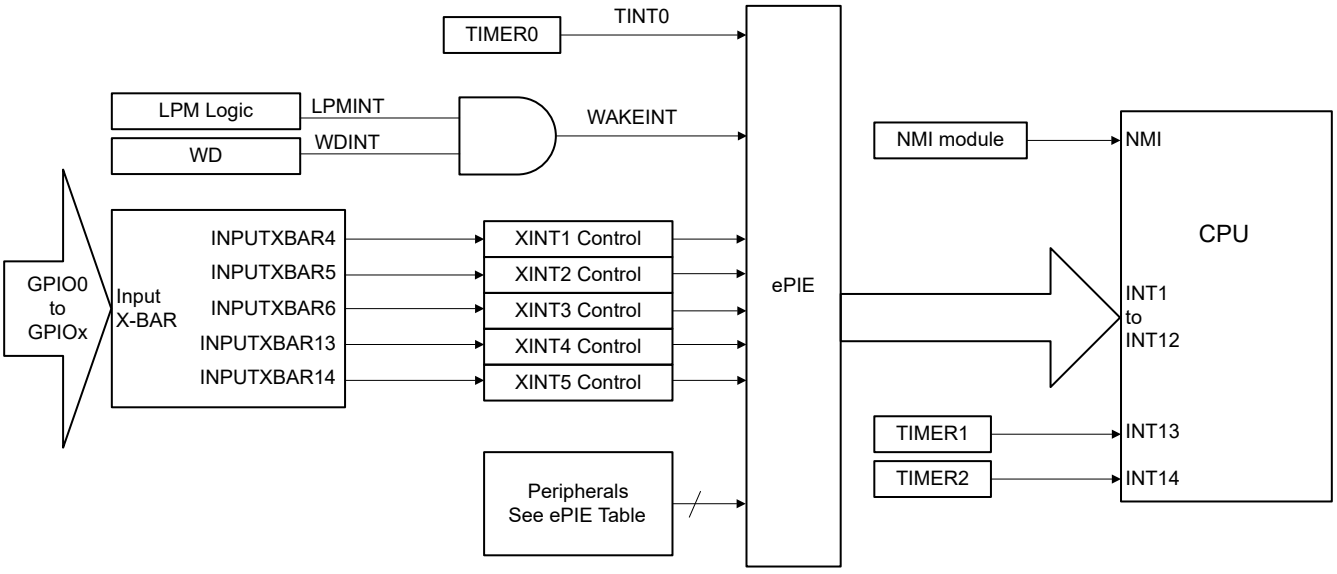


Figure 3-1. Device Interrupt Architecture

3.5.2.1 Peripheral Stage

Each peripheral has a unique interrupt configuration, which is described in that peripheral's chapter. Some peripherals allow multiple events to trigger the same interrupt signal. For example, a communications peripheral can use the same interrupt to indicate that data has been received or that there has been a transmission error. The cause of the interrupt can be determined by reading the peripheral's status register. Often, the bits in the status register must be cleared manually before another interrupt is generated.

3.5.2.2 PIE Stage

The PIE provides individual flag and enable register bits for each of the peripheral interrupt signals, which are sometimes called PIE channels. These channels are grouped according to the associated CPU interrupt. Each PIE group has one 4-bit enable register (PIEIERx), one 4-bit flag register (PIEIFRx) , and one bit in the PIE acknowledge register (PIEACK). The PIEACK register bit acts as a common interrupt mask for the entire PIE group.

When the CPU receives an interrupt, the CPU fetches the address of the ISR from the PIE. The PIE returns the vector for the lowest-numbered channel in the group that is both flagged and enabled. This gives lower-numbered interrupts a higher priority when multiple interrupts are pending.

If no interrupt is both flagged and enabled, the PIE returns the vector for channel 1. This condition does not happen unless software changes the state of the PIE while an interrupt is propagating. Section 3.5.4 contains procedures for safely modifying the PIE configuration once interrupts have been enabled.

3.5.2.3 CPU Stage

Like the PIE, the CPU provides flag and enable register bits for each of the interrupts. There is one enable register (IER) and one flag register (IFR), both of which are internal CPU registers. There is also a global interrupt mask, which is controlled by the INTM bit in the ST1 register. This mask can be set and cleared using the CPU SETC and CLRC instructions. In C code, C2000Ware's DINT and EINT macros can be used for this purpose.

Writes to IER and INTM are atomic operations. In particular, if INTM is set, the next instruction in the pipeline runs with interrupts disabled. No software delays are needed.

3.5.3 Interrupt Entry Sequence

Figure 3-2 shows how peripheral interrupts propagate to the CPU.

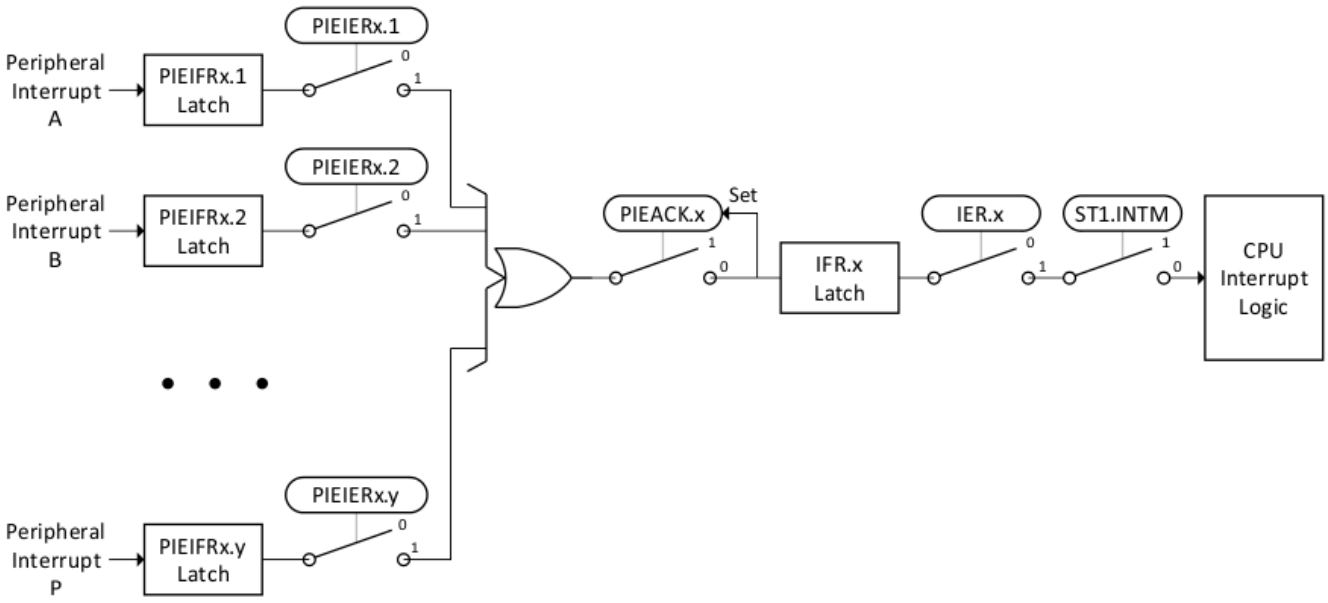


Figure 3-2. Interrupt Propagation Path

When a peripheral generates an interrupt (on PIE group x, channel y), the interrupt triggers the following sequence of events:

1. The interrupt is latched in PIEIFRx.y.
2. If PIEIERx.y is set, the interrupt propagates.
3. If PIEACK.x is clear, the interrupt propagates and PIEACK.x is set.
4. The interrupt is latched in IFR.x.
5. If IER.x is set, the interrupt propagates.
6. If INTM is clear, the CPU receives the interrupt.
7. Any instructions in the D2 or later stage of the pipeline are run to completion. Instructions in earlier stages are flushed.
8. The CPU saves the context on the stack.
9. IFR.x and IER.x are cleared. INTM is set. EALLOW is cleared.
10. The CPU fetches the ISR vector from the PIE. PIEIFRx.y is cleared.
11. The CPU branches to the ISR.

The interrupt latency is the time between PIEIFRx.y latching the interrupt and the first ISR instruction entering the execution stage of the CPU pipeline. The minimum interrupt latency is 14 SYSCLK cycles. Wait states on the ISR or stack memories add to the latency. External interrupts add a minimum of 2 SYSCLK cycles for GPIO synchronization plus extra time for input qualification (if used). Loops created using the C28x RPT instruction cannot be interrupted.

3.5.4 Configuring and Using Interrupts

At power-up, no interrupts are enabled by default. The PIEIER and IER registers are cleared and INTM is set. The application code is responsible for configuring and enabling all peripheral interrupts.

3.5.4.1 Enabling Interrupts

To enable a peripheral interrupt, perform the following steps:

1. Disable interrupts globally (DINT or SETC INTM).
2. Enable the PIE by setting the ENPIE bit of the PIECTRL register.
3. Write the ISR vector for each interrupt to the appropriate location in the PIE vector table, which can be found in Section 3.5.8. Note that the vector table is EALLOW-protected.
4. Set the appropriate PIEIERx bit for each interrupt. The PIE group and channel assignments can be found in Section 3.5.8.
5. Set the CPU IER bit for any PIE group containing enabled interrupts.
6. Enable the interrupt in the peripheral.
7. Enable interrupts globally (EINT or CLRC INTM).

Step 4 does not apply to the Timer1 and Timer2 interrupts, which connect directly to the CPU.

3.5.4.2 Handling Interrupts

ISRs are similar to normal functions, but must do the following:

1. Save and restore the state of certain CPU registers (if used).
2. Clear the PIEACK bit for the interrupt group.
3. Return using the IRET instruction.

Requirements 1 and 3 are handled automatically by the TMS320C28x C compiler if the function is defined using the __interrupt keyword. For information on this keyword, see the Keywords section of the *TMS320C28x Optimizing C/C++ Compiler v6.2.4 User's Guide*. For information on writing assembly code to handle interrupts, see the Standard Operation for Maskable Interrupts section of the *TMS320C28x CPU and Instruction Set Reference Guide*.

The PIEACK bit for the interrupt group must be cleared manually in user code. This is normally done at the end of the ISR. If the PIEACK bit is not cleared, the CPU does not receive any further interrupts from that group. This does not apply to the Timer1 and Timer2 interrupts, which do not go through the PIE.

3.5.4.3 Disabling Interrupts

To disable all interrupts, set the CPU global interrupt mask using DINT or SETC INTM. It is not necessary to add NOPs after setting INTM or modifying IER – the next instruction executes with interrupts disabled.

Individual interrupts can be disabled using the PIEIERx registers, but care must be taken to avoid race conditions. If an interrupt signal is already propagating when the PIEIER write completes, the signal can reach the CPU and trigger a spurious interrupt condition. To avoid this, use the following procedure:

1. Disable interrupts globally (DINT or SETC INTM).
2. Clear the PIEIER bit for the interrupt.
3. Wait 5 cycles to make sure that any propagating interrupt has reached the CPU IFR register.
4. Clear the CPU IFR bit for the interrupt's PIE group.
5. Clear the PIEACK bit for the interrupt's PIE group.
6. Enable interrupts globally (EINT or CLRC INTM).

Interrupt groups can be disabled using the CPU IER register. This cannot cause a race condition, so no special procedure is needed.

PIEIFR bits must never be cleared in software since the read/modify/write operation can cause incoming interrupts to be lost. The only safe way to clear a PIEIFR bit is to have the CPU take the interrupt. The following procedure can be used to bypass the normal ISR:

1. Disable interrupts globally (DINT or SETC INTM).
2. Modify the PIE vector table to map the PIEIFR bit interrupt vector to an empty ISR. This ISR only contains a return from interrupt (IRET) instruction.
3. Disable the interrupt in the peripheral registers.
4. Enable interrupts globally (EINT or CLRC INTM).
5. Wait for the pending interrupt to be serviced by the empty ISR.
6. Disable interrupts globally.
7. Modify the PIE vector table to map the interrupt vector back to the original ISR.
8. Clear the PIEACK bit for the interrupt's PIE group.
9. Enable interrupts globally.

3.5.4.4 Nesting Interrupts

By default, interrupts do not nest. It is possible to nest and prioritize interrupts using software control of the IER and PIEIERx registers. Example code can be found in C2000Ware and documentation is available at software-dl.ti.com/C2000/docs/c28x_interrupt_nesting/html/index.html.

3.5.4.5 Vector Address Validity Check

The ePIE vector table memory is protected using a parity check. Upon each vector fetch from the ePIE, a parity check is performed. If a parity failure occurs during vector fetch, the ePIE returns either a user defined error handler routine (if PIEVERRADDR is defined with a non 0x003F FFFF value), or the default boot ROM handler at address 0x3F FFBE. The ePIE also sends trip signals to the MCPWMs.

The parity check only returns the error handler value if the failure occurs during vector fetch. Parity errors during a data read are handled by the memory controller module and logged by the UCERRFLG register in the MEMORY_ERROR_REGS. The address that caused the error is located in the UCCPUREADDR register. If the error address logged is between 0xD00 to 0xDA0, then the error is a PIE parity error. Additionally, a parity error during a vector fetch does not flag an uncorrectable error NMI.

3.5.5 PIE Channel Mapping

Table 3-3 shows the PIE group and channel assignments for each peripheral interrupt. Each row is a group, and each column is a channel within that group. When multiple interrupts are pending, the lowest-numbered channel is the lowest-numbered group is serviced first. Thus, the interrupts at the top of the table have the highest priority, and the interrupts at the bottom have the lowest priority.

Table 3-3. PIE Channel Mapping

GROUP.CHANNEL	Interrupt Source
INT1.1	ADCA1
INT1.2	
INT1.3	XINT1
INT1.4	XINT2
INT2.1	SYS_ERR
INT2.2	TIMER0
INT2.3	WAKE
INT2.4	PWM1
INT3.1	
INT3.2	PWM3
INT3.3	
INT3.4	
INT4.1	
INT4.2	Reserved
INT4.3	ECAP1
INT4.4	
INT5.1	EQEP1
INT5.2	SPIA_RX
INT5.3	SPIA_TX
INT5.4	
INT6.1	
INT6.2	DCC0
INT6.3	DMA_CH1
INT6.4	DMA_CH2
INT7.1	
INT7.2	I2CA
INT7.3	I2CA_FIFO
INT7.4	SCIA_RX
INT8.1	SCIA_TX
INT8.2	SCIB_RX
INT8.3	SCIB_TX
INT8.4	
INT9.1	
INT9.2	UART0_INT
INT9.3	ADCA_EVT
INT9.4	ADCA2
INT10.1	
INT10.2	
INT10.3	XINT3
INT10.4	XINT4

Table 3-3. PIE Channel Mapping (continued)

GROUP.CHANNEL	Interrupt Source
INT11.1	XINT5
INT11.2	FLSS_INT
INT11.3	
INT11.4	
INT12.1	
INT12.2	
INT12.3	
INT12.4	Reserved

3.5.6 PIE Interrupt Priority

3.5.6.1 Channel Priority

For every PIE group, the low-number channels in the group have the highest priority. For instance in PIE group 1, channel 1.1 has priority over channel 1.3. If those two enabled interrupts occurred simultaneously, channel 1.1 is serviced first with channel 1.3 left pending. Once the ISR for channel 1.1 completes and provided there are no other enabled and pending interrupts for PIE group 1, channel 1.3 is serviced. However, for the CPU to service any more interrupts from a PIE group, PIEACK for the group must be cleared. For this specific example, for channel 1.3 to be serviced, the channel 1.1 ISR has to clear PIEACK for group 1.

The following example describes an alternative scenario: channel 1.1 is currently being serviced by the CPU, channel 1.3 is pending and before the channel 1.1 ISR completes, channel 1.2 that is enabled also comes in. Since channel 1.2 has a higher priority than channel 1.3, the CPU services channel 1.2 and channel 1.3 is still left pending. Using the steps from the Interrupt Entry Sequence (Section 3.5.3), a channel 1.2 interrupt can happen as late as step 10 (The CPU fetches the ISR vector from the PIE. PIEIFRx.y is cleared) and the interrupt is still serviced ahead of channel 1.3.

3.5.6.2 Group Priority

Generally, the lowest channel in the lowest PIE group has the highest priority. An example of this is channels 1.1 and 2.1. Those two channels have the highest priority in the respective groups. If the interrupts for those two enabled channels happened simultaneously and provided there are no other enabled and pending interrupts, channel 1.1 is serviced first by the CPU with channel 2.1 left pending.

However, there are cases where channel priority supersedes group priority. This special case happens depending on which step the CPU is currently at in the Interrupt Entry Sequence (Section 3.5.3).

The following illustrates an example of this special case.

The CPU is about to service channel 2.3 and is currently going through the steps in the Interrupt Entry Sequence (Section 3.5.3).

1. As the CPU reaches step 10 (The CPU fetches the ISR vector from the PIE. PIEIFRx.y is cleared), two enabled interrupts: channel 1.1 and channel 2.1 come in.
2. Due to channel priority, channel 2.1 is serviced ahead of channel 2.3. However, group priority dictates that channel 1.1 be serviced ahead of channels 2.1 and 2.3.
3. Channel priority supersedes here and channel 2.1 is serviced ahead of channels 1.1 and 2.3.
4. After channel 2.1 completes, channel 1.1 is serviced followed by channel 2.3.

Group priority is only maintained if no interrupts are currently being serviced, that is, the Interrupt Entry Sequence (Section 3.5.3) is not executing.

3.5.7 System Error

SYS_ERR consolidates several sources of interrupts (see Figure 3-3). These sources set the respective bit in the SYS_ERR_INT_FLG register. Any set bit in the SYS_ERR_INT_FLG register also sets the global interrupt (GINT) bit. The GINT bit has to be cleared before any SYS_ERR interrupt is generated. If the GINT bit is cleared with the source flags still set, another SYS_ERR interrupt is signaled; therefore, clear the source flags before clearing the GINT bit. For System Error sources, please refer to SYS_ERR_INT_FLG register.

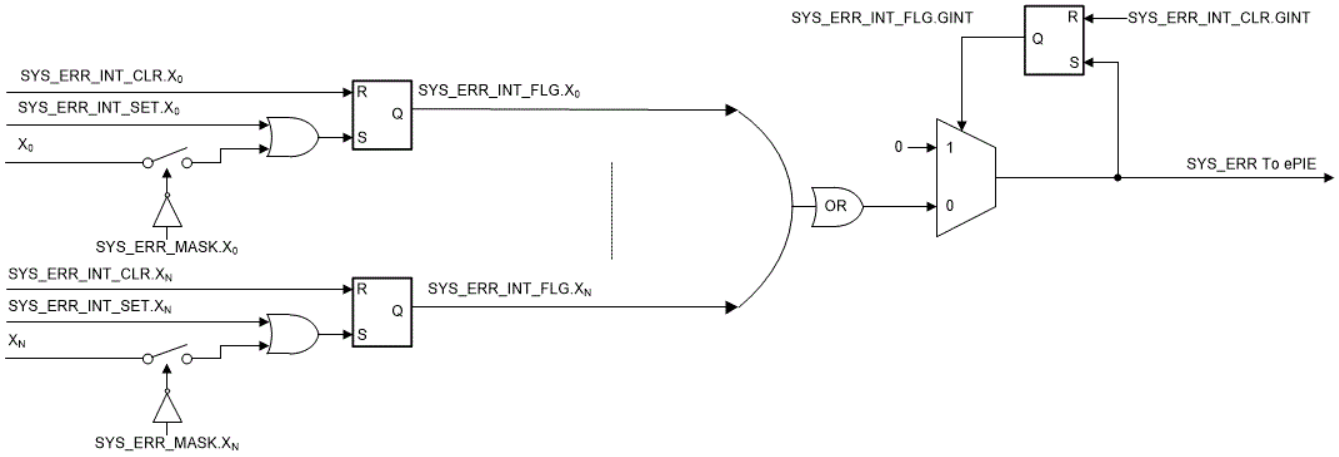


Figure 3-3. System Error

3.5.8 Vector Tables

Table 3-4 shows the CPU interrupt vector table. The vectors for INT1 – INT12 are not used in this device. The reset vector is fetched from the boot ROM instead of from this table. All vectors are EALLOW-protected.

Table 3-5 shows the PIE vector table.

Table 3-4. CPU Interrupt Vectors

Name	Vector ID	Address	Size (x16)	Description	Core Priority	ePIE Group Priority
Reset	0	0x0000 0D00	2	Reset is always fetched from location 0x003F_FFC0 in Boot ROM	1 (Highest)	-
INT1	1	0x0000 0D02	2	Not used. See PIE Group 1	5	-
INT2	2	0x0000 0D04	2	Not used. See PIE Group 2	6	-
INT3	3	0x0000 0D06	2	Not used. See PIE Group 3	7	-
INT4	4	0x0000 0D08	2	Not used. See PIE Group 4	8	-
INT5	5	0x0000 0D0A	2	Not used. See PIE Group 5	9	-
INT6	6	0x0000 0D0C	2	Not used. See PIE Group 6	10	-
INT7	7	0x0000 0D0E	2	Not used. See PIE Group 7	11	-
INT8	8	0x0000 0D10	2	Not used. See PIE Group 8	12	-
INT9	9	0x0000 0D12	2	Not used. See PIE Group 9	13	-
INT10	10	0x0000 0D14	2	Not used. See PIE Group 10	14	-
INT11	11	0x0000 0D16	2	Not used. See PIE Group 11	15	-
INT12	12	0x0000 0D18	2	Not used. See PIE Group 12	16	-
INT13	13	0x0000 0D1A	2	CPU TIMER1 Interrupt	17	-
INT14	14	0x0000 0D1C	2	CPU TIMER2 Interrupt	18	-
DATALOG	15	0x0000 0D1E	2	CPU Data Logging Interrupt	19 (lowest)	-
RTOSINT	16	0x0000 0D20	2	CPU Real-Time OS Interrupt	4	-
RSVD	17	0x0000 0D22	2	Reserved	2	-
NMI	18	0x0000 0D24	2	Non-Maskable Interrupt	3	-
ILLEGAL	19	0x0000 0D26	2	Illegal Instruction (ITRAP)	-	-
USER 1	20	0x0000 0D28	2	User-Defined Trap	-	-
USER 2	21	0x0000 0D2A	2	User-Defined Trap	-	-
USER 3	22	0x0000 0D2C	2	User-Defined Trap	-	-
USER 4	23	0x0000 0D2E	2	User-Defined Trap	-	-
USER 5	24	0x0000 0D30	2	User-Defined Trap	-	-
USER 6	25	0x0000 0D32	2	User-Defined Trap	-	-
USER 7	26	0x0000 0D34	2	User-Defined Trap	-	-
USER 8	27	0x0000 0D36	2	User-Defined Trap	-	-
USER 9	28	0x0000 0D38	2	User-Defined Trap	-	-
USER 10	29	0x0000 0D3A	2	User-Defined Trap	-	-
USER 11	30	0x0000 0D3C	2	User-Defined Trap	-	-
USER 12	31	0x0000 0D3E	2	User-Defined Trap	-	-

Table 3-5. PIE Interrupt Vectors

Name	Vector ID	Address	Size (x16)	Description	Core Priority	ePIE Group priority
PIE Group 1 Vectors - Muxed into CPU INT1						
INT1.1	32	0x0000 0D40	2	ADCA1 interrupt	5	1 (Highest)
INT1.2	33	0x0000 0D42	2	Reserved	5	2
INT1.3	34	0x0000 0D44	2	XINT1 interrupt	5	3
INT1.4	35	0x0000 0D46	2	XINT2 interrupt	5	4
PIE Group 2 Vectors - Muxed into CPU INT2						
INT2.1	36	0x0000 0D48	2	SYS_ERR interrupt	6	1 (Highest)
INT2.2	37	0x0000 0D4A	2	TIMER0 interrupt	6	2
INT2.3	38	0x0000 0D4C	2	WAKE interrupt	6	3
INT2.4	39	0x0000 0D4E	2	MCPWM1 interrupt	6	4
PIE Group 3 Vectors - Muxed into CPU INT3						
INT3.1	40	0x0000 0D50	2	Reserved	7	1 (Highest)
INT3.2	41	0x0000 0D52	2	MCPWM3 interrupt	7	2
INT3.3	42	0x0000 0D54	2	Reserved	7	3
INT3.4	43	0x0000 0D56	2	Reserved	7	4
PIE Group 4 Vectors - Muxed into CPU INT4						
INT4.1	44	0x0000 0D58	2	Reserved	8	1 (Highest)
INT4.2	45	0x0000 0D5A	2	Reserved	8	2
INT4.3	46	0x0000 0D5C	2	ECAP1 interrupt	8	3
INT4.4	47	0x0000 0D5E	2	Reserved	8	4
PIE Group 5 Vectors - Muxed into CPU INT5						
INT5.1	48	0x0000 0D60	2	EQEP1 interrupt	9	1 (Highest)
INT5.2	49	0x0000 0D62	2	SPIA_RX interrupt	9	2
INT5.3	50	0x0000 0D64	2	SPIA_TX interrupt	9	3
INT5.4	51	0x0000 0D66	2	Reserved	9	4
PIE Group 6 Vectors - Muxed into CPU INT6						
INT6.1	52	0x0000 0D68	2	Reserved	10	1 (Highest)
INT6.2	53	0x0000 0D6A	2	DCC0 interrupt	10	2
INT6.3	54	0x0000 0D6C	2	DMA_CH1 interrupt	10	3
INT6.4	55	0x0000 0D6E	2	DMA_CH2 interrupt	10	4
PIE Group 7 Vectors - Muxed into CPU INT7						
INT7.1	56	0x0000 0D70	2	Reserved	11	1 (Highest)
INT7.2	57	0x0000 0D72	2	I2CA interrupt	11	2
INT7.3	58	0x0000 0D74	2	I2CA_FIFO interrupt	11	3
INT7.4	59	0x0000 0D76	2	SCIA_RX interrupt	11	4
PIE Group 8 Vectors - Muxed into CPU INT8						
INT8.1	60	0x0000 0D78	2	SCIA_TX interrupt	12	1 (Highest)
INT8.2	61	0x0000 0D7A	2	SCIB_RX interrupt	12	2
INT8.3	62	0x0000 0D7C	2	SCIB_TX interrupt	12	3
INT8.4	63	0x0000 0D7E	2	Reserved	12	4

Table 3-5. PIE Interrupt Vectors (continued)

Name	Vector ID	Address	Size (x16)	Description	Core Priority	ePIE Group priority
PIE Group 9 Vectors - Muxed into CPU INT9						
INT9.1	64	0x0000 0D80	2	Reserved	13	1 (Highest)
INT9.2	65	0x0000 0D82	2	UARTA interrupt	13	2
INT9.3	66	0x0000 0D84	2	ADCA Event interrupt	13	3
INT9.4	67	0x0000 0D86	2	ADCA2 interrupt	13	4
PIE Group 10 Vectors - Muxed into CPU INT10						
INT10.1	68	0x0000 0D88	2	Reserved	14	1 (Highest)
INT10.2	69	0x0000 0D8A	2	Reserved	14	2
INT10.3	70	0x0000 0D8C	2	XINT3 interrupt	14	3
INT10.4	71	0x0000 0D8E	2	XINT4 interrupt	14	4
PIE Group 11 Vectors - Muxed into CPU INT11						
INT11.1	72	0x0000 0D90	2	XINT5 interrupt	15	1 (Highest)
INT11.2	73	0x0000 0D92	2	FLSS interrupt	15	2
INT11.3	74	0x0000 0D94	2	Reserved	15	3
INT11.4	75	0x0000 0D96	2	Reserved	15	4
PIE Group 12 Vectors - Muxed into CPU INT12						
INT12.1	76	0x0000 0D98	2	Reserved	16	1 (Highest)
INT12.2	77	0x0000 0D9A	2	Reserved	16	2
INT12.3	78	0x0000 0D9C	2	Reserved	16	3
INT12.4	79	0x0000 0D9E	2	Reserved	16	4

3.6 Exceptions and Non-Maskable Interrupts

This section describes system-level error conditions that can trigger a non-maskable interrupt (NMI). The interrupt allows the application to respond to the error.

3.6.1 Configuring and Using NMIs

An incoming NMI sets a status bit in the NMIFLG register and starts the NMI watchdog counter. This counter is clocked by the SYSCLK, and if the counter reaches the value in the NMIWDPRD register, the counter triggers an NMI watchdog reset (NMIWDRS). To prevent this, the NMI handler must clear the flag bit using the NMIFLGCLR register. Once all flag bits are clear, the NMIINT bit in the NMIFLG register can also be cleared to allow future NMIs to be taken.

The NMI module is enabled by the boot ROM during the startup process. To respond to NMIs, an NMI handler vector must be written to the PIE vector table.

3.6.2 Emulation Considerations

The NMI watchdog counter behaves as follows under debug conditions:

CPU Suspended	When the CPU is suspended, the NMI watchdog counter is suspended.
Run-Free Mode	When the CPU is placed in run-free mode, the NMI watchdog counter resumes operation as normal.
Real-Time Single-Step Mode	When the CPU is in real-time single-step mode, the NMI watchdog counter is suspended. The counter remains suspended even within real-time interrupts.
Real-Time Run-Free Mode	When the CPU is in real-time run-free mode, the NMI watchdog counter operates as normal.

3.6.3 NMI Sources

There are several types of hardware errors that can trigger an NMI. Additional information about the error is usually available from the module that detects the error.

3.6.3.1 Missing Clock Detection Logic

The missing clock detection logic monitors OSCCLK for failure. If the OSCCLK source stops, the PLL is bypassed, OSCCLK is connected to SECCLK, and an NMI is fired to the CPU. For more information on missing clock detection, see Section 3.7.12.1.

3.6.3.2 Flash Uncorrectable ECC Error

A double-bit ECC data error or single-bit ECC address error in a Flash read triggers an NMI. Single-bit ECC data errors do not trigger an NMI, but can optionally trigger a normal peripheral interrupt.

3.6.3.3 Software-Forced Error

There is a special NMI source that can only be triggered by writing to the SWERR bit in the NMIFLGFRC register. Since the SWERR flag is never set by a real hardware fail, the flag can be used to implement a self-test mode for the NMI subsystem.

3.6.4 Illegal Instruction Trap (ITRAP)

If the CPU tries to execute an illegal instruction, the CPU generates a special interrupt called an illegal instruction trap (ITRAP). This interrupt is non-maskable and has a vector in the PIE vector table. For more information about ITRAPs, see the Illegal-Instruction Trap section of the *TMS320C28x DSP CPU and Instruction Set Reference Guide*.

Note

A RAM fetch access violation triggers an ITRAP in addition to the normal peripheral interrupt for RAM access violations. The CPU handles the ITRAP first.

3.6.5 ERRORSTS Pin

A signal called ERRORSTS can be output to GPIO24, GPIO28, or GPIO29. This signal goes low when any bit is set in the NMI shadow flag register (NMISHDFLG). The signal can be used to alert an external system to a problem in the microcontroller. Since the state of ERRORSTS is based on the shadow flags, ERRORSTS remains low until the flags are cleared by the CPU or a power-on reset occurs.

All GPIO pins are inputs on power-up. If the state of the chosen ERRORSTS pin during power-up is important, an external pull-down must be connected to the pin.

The ERRORSTS pin is an 'always output' pin and remains high until an error is detected inside the chip. On an error, the ERRORSTS pin goes low (default polarity) until the corresponding internal error status flag for that error source is cleared. Figure 3-4 shows the functionality of the ERRORSTS pin.

The ERRORSTS pin is tri-stated until the chip power rails ramp up to the lower operational limit. As the ERRORSTS pin is an active-low pin (default polarity), users who care about the state of this pin during power-up can connect an external pull-down on this pin.

Following enhancement has been made on this device for ERRORSTS pin logic:

- Polarity of Error pin has been made configurable through the ERRORCTL register (default setting is active-low polarity).
- To enable testing of the Error pin, capability to force and clear the Error pin from software has been provided.
- Additional sources of Error have been added to ERRORSTS:
  - CPU1 Watchdog reset
  - Error on a PIE vector fetch

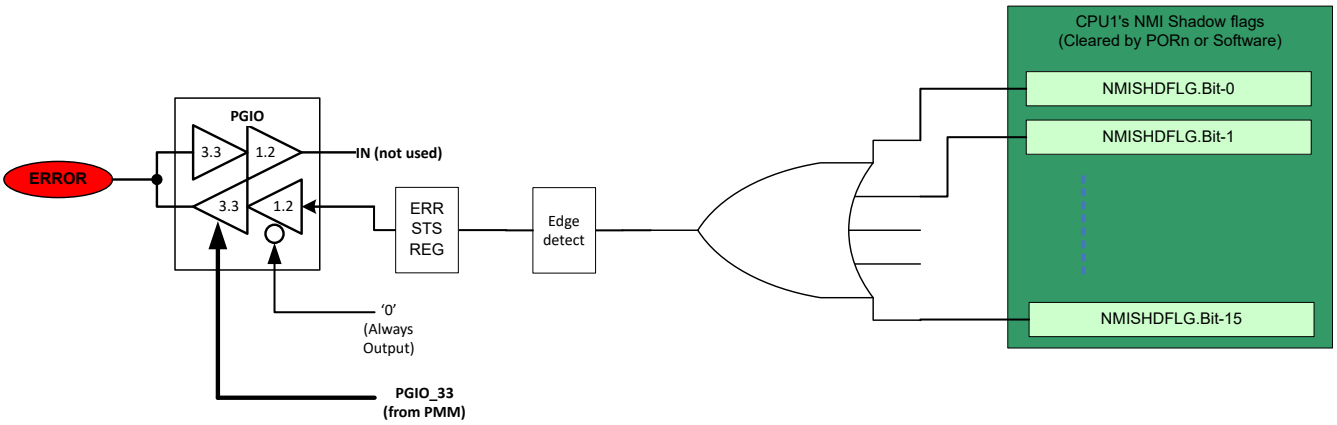


Figure 3-4. ERRORSTS Pin Diagram

3.7 Clocking

This section explains the clock sources and clock domains on this device, and how to configure them for application use. Figure 3-5 and Figure 3-6 provide an overview of the device's clocking system. The relation between PLLRAWCLK and OSCCLK is given in Equation 1.

$$f_{PLLRAWCLK} = \frac{f_{OSCCLK} \times (\text{QDIV})}{(\text{PDIV}) \times (\text{RDIVCLK0})} \quad (1)$$

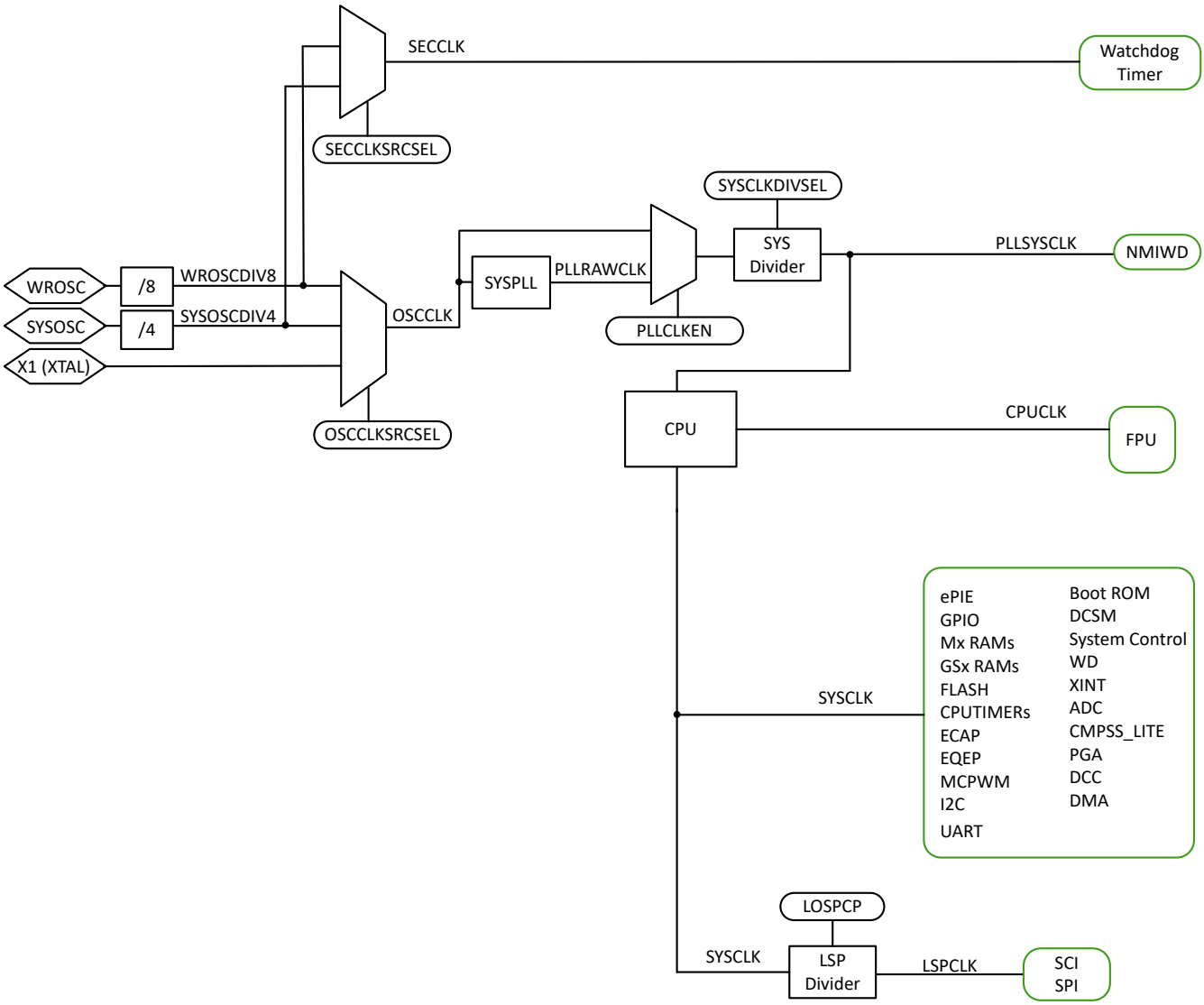


Figure 3-5. Clocking System

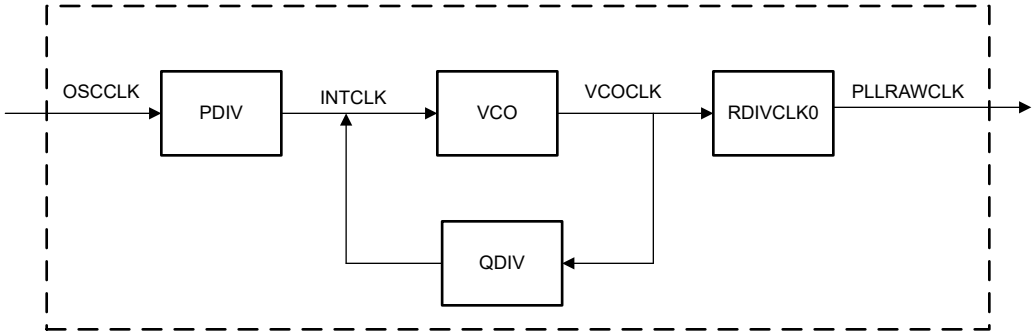


Figure 3-6. System PLL

3.7.1 Clock Sources

All of the clocks in the device are derived from one of four clock sources.

3.7.1.1 Primary Internal Oscillator (SYSOSC)

At power-up, the device is clocked from an on-chip 32MHz oscillator (SYSOSC). SYSOSC (converted to SYSOSCDIV4 after passing through a fixed divby4 divider) is the primary internal clock source and is the default system clock at reset. SYSOSC is used to run the boot ROM and can be used as the system clock source for the application.

3.7.1.2 Backup Wide-Range Oscillator (WROSC)

The device also includes a redundant wide-range on-chip oscillator (WROSC). WROSC (converted to WROSCDIV8 after passing through a fixed divby divider) is a backup clock source with output frequency of 20-70MHz that normally only clocks the watchdog timers and missing clock detection circuit (MCD). If MCD is enabled and a missing system clock is detected, the system PLL is bypassed and all system clocks are connected to WROSC automatically. WROSC can also be manually selected as the system clock source for debug purposes. Note that WROSC must not be used as a source for PLL.

3.7.1.3 External Oscillator (XTAL)

The device supports an external clock source (XTAL), which can be used as the main system. Frequency limits and timing requirements are found in the *F28E12x Real-Time Microcontrollers Data Sheet*. External clock sources use the X1/GPIO19 and X2/GPIO18 pins. After power-up, the X1 and X2 pin functionality can be enabled by following the procedure in Section 3.7.6.

Three types of external clock sources are supported:

- A single-ended 3.3V external clock. The clock signal must be connected to X1, as shown in Figure 3-7.

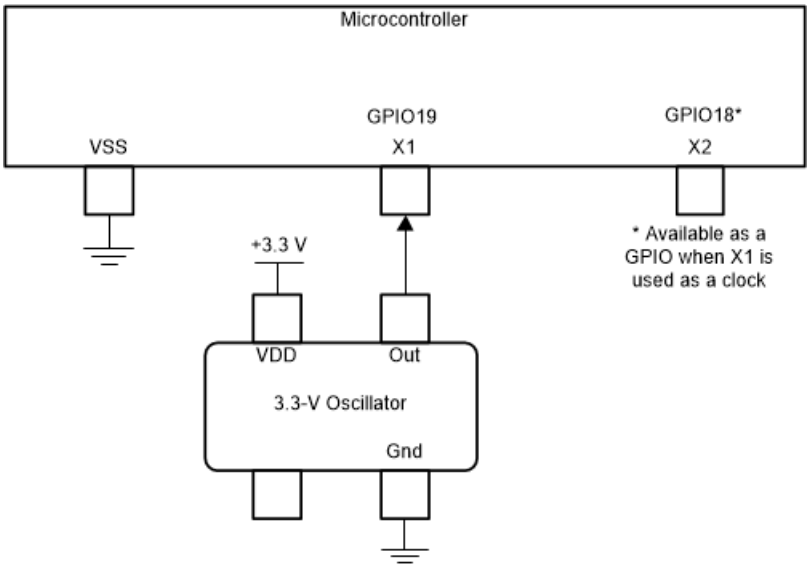


Figure 3-7. Single-ended 3.3V External Clock

- An external crystal. The crystal must be connected across X1 and X2 with the load capacitors connected to VSS, as shown in Figure 3-8.

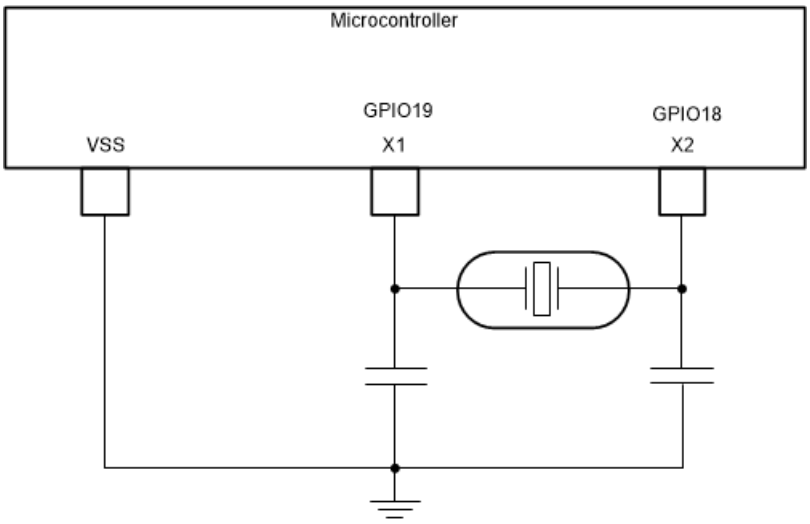


Figure 3-8. External Crystal

- An external resonator. The resonator must be connected across X1 and X2 with the ground connected to VSS, as shown in Figure 3-9.

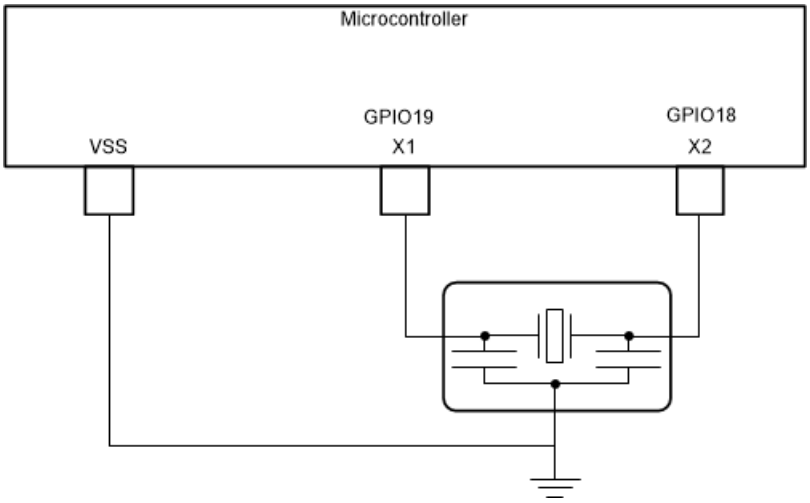


Figure 3-9. External Resonator

Table 3-6. ALT Modes

XTALCR Bit[1]		Operating Mode	GPIO19 Available on X1?	GPIO18 Available on X2?
OSCOFF	SE			
0	0	Crystal Mode: Quartz crystal connected to X1/X2	No	No
0	1	Single-Ended Mode: External clock on X1	No	Yes
1	0	Oscillator off	Yes	Yes
1	1	Single-Ended Mode: External clock on X1[2]	No	Yes

(1) OSCOFF and SE determine the ALT mode of GPIO18 and GPIO19.
(2) There is an approximately 1Kohm pull-down on X1 in this mode, external single-ended clock must be able to drive this load.

3.7.2 Derived Clocks

The clock sources discussed in the previous section can be multiplied (using PLL) and divided down to produce the desired clock frequencies for the application. This process produces a set of derived clocks, which are described in this section.

3.7.2.1 Oscillator Clock (OSCCLK)

One of SYSOSC, XTAL, or WROSC must be chosen to be the controller reference clock (OSCCLK) for the CPU and most of the peripherals. OSCCLK can be used directly or applied through the system PLL to reach a higher frequency. At reset, OSCCLK is the default system clock and is connected to SYSOSC.

3.7.2.2 System PLL Output Clock (PLLRAWCLK)

The system PLL allows the device to run at the maximum rated operating frequency, and in most applications generates the main system clock. This PLL uses OSCCLK as a reference. PLLRAWCLK is the output of the PLL voltage-controlled oscillator (VCO). For configuration instructions, see Section 3.7.6.

3.7.3 Device Clock Domains

The device clock domains supply the clock inputs of the various modules in the device. The inputs are connected to the derived clocks, either directly or through an additional divider.

3.7.3.1 System Clock (PLLSYSCLK)

The NMI watchdog timer has a clock domain (PLLSYSCLK). Despite the name, PLLSYSCLK can be connected to the system PLL (PLLRAWCLK) or to OSCCLK. The chosen clock source is run through a frequency divider, which is configured using the SYSCLKDIVSEL register. PLLSYSCLK is gated in HALT mode.

3.7.3.2 CPU Clock (CPUCLK)

The CPU has a clock (CPUCLK) that is used to clock the CPU and Flash wrapper. This clock is identical to PLLSYSCLK, but is gated when the CPU enters IDLE, STANDBY, or HALT mode.

3.7.3.3 CPU Subsystem Clock (SYSCLK)

The CPU provides a clock (SYSCLK) to the private (M0 and M1), global shared (GS0), boot ROM and other peripherals. This clock is identical to PLLSYSCLK, but is gated when the CPU enters HALT or STANDBY mode.

3.7.3.4 Low-Speed Peripheral Clock (LSPCLK and PERx.LSPCLK)

The SCI and SPI modules can communicate at bit rates that are much slower than the CPU frequency. These modules are connected to a shared clock divider, which generates a low-speed peripheral clock (LSPCLK) derived from SYSCLK. LSPCLK uses a /4 divider by default, but the ratio can be changed using the LOSPCP register.

3.7.3.5 CPU Timer2 Clock (TIMER2CLK)

CPU timers 0 and 1 are connected to SYSCLK. Timer 2 is connected to SYSCLK by default, but can also be connected to SYSOSC, WROSC, or XTAL using the TMR2CLKCTL register. This register also provides a separate prescale divider for timer 2. If a non-SYSCLK source is used, the source must be divided down to no more than half the SYSCLK frequency.

The main reason to use a non-SYSCLK source is for internal frequency measurement. In most applications, timer 2 runs off of SYSCLK.

3.7.4 XCLKOUT

It is sometimes necessary to observe a clock directly for debug and testing purposes. The external clock output (XCLKOUT) feature supports this by connecting a clock to an external pin, which can be GPIO16 or GPIO18. The available clock sources are PLLSYSCLK, PLLRAWCLK, SYSCLK, SYSOSC, WROSC, and XTAL.

To use XCLKOUT, first select the clock source using the CLKSRCCTL3 register. Next, select the desired output divider using the XCLKOUTDIVSEL register. Finally, connect GPIO16 or GPIO18 to mux channel 11 using the GPIO configuration registers.

3.7.5 Clock Connectivity

Table 3-7 shows the clock connections sorted by the clock domain and Table 3-8 shows the clock connections sorted by the module name.

Table 3-7. Clock Connections Sorted by Clock Domain

Clock Domain	Module Name
CPUCLK	FPU
SYSCLK	ePIE
	GSx RAMs
	Flash
	Boot ROM
	GPIO Input Sync and Qual
	WD
	XINT
	Timer0 - 2
	DCC0
	MCPWM1-2
	eCAP1
	eQEP1
	ADCA
	CMPSS_LITE1 - 3
	I2CA
	DCSM
	UART
PLLSYSCLK	CPU
	NMIWD
PERx.LSPCLK	SCIA - B
	SPIA
WDCLK (SECCLK)	Watchdog Timer

Table 3-8. Clock Connections Sorted by Module Name

Module Name	Clock Domain
ADCA	PERx.SYSCLK
Boot ROM	SYSCLK
CMPSS_LITE (1 - 3)	PERx.SYSCLK
CPU	PLLSYSCLK
CPU Timers (0 - 2)	PERx.SYSCLK
DCC0	PERx.SYSCLK
DCSM	SYSCLK
eCAP1	PERx.SYSCLK
ePIE	SYSCLK
MCPWM1 & MCPWM3	PERx.SYSCLK
eQEP1	PERx.SYSCLK
Flash	SYSCLK
FPU	CPUCLK
GPIO Input Sync and Qual	SYSCLK
I2CA	PERx.SYSCLK
Mx RAMs	SYSCLK
NMIWD	PLLSYSCLK
SCI (A - B)	PERx.LSPCLK
SPIA	PERx.LSPCLK
UART	PERx.SYSCLK
Watchdog Timer	WDCLK (SECOSC)

3.7.6 Clock Source and PLL Setup

The needs of the application are what ultimately determine the clock configuration. Specific concerns such as application performance, power consumption, total system cost, and EMC are beyond the scope of this document, but can provide answers to the following questions:

1. What is the desired CPU frequency?
2. What types of external oscillators or clock sources are available?

3.7.7 Using an External Crystal or Resonator

The X1 and X2 pins double as GPIO19 and GPIO18. At power-up, these pins are in GPIO mode and the on-chip crystal oscillator is powered off. The following procedure can be used to switch the pins to X1 and X2 mode and enable the oscillator:

1. Clear the XTALCR.OSCOFF bit.
2. Wait for the crystal to power up (1ms is the typical wait time but this depends on the crystal that is being used).
3. Clear the X1 counter by writing a 1 to X1CNT.CLR and keep clearing until the X1 counter value in the X1CNT register is no longer saturated 2047 (0x7FF).
4. Wait for the X1 counter value in the X1CNT register to reach 2047 (0x7FF). Repeat steps 3 and 4 three additional times.
5. Select XTAL as the OSCCLK source by writing a 1 to CLKSRCCTL1.OSCCLKSRCSEL.
6. Check the MCLKSTS bit in the MCDCR register. If the bit set, the oscillator has not finished powering up, and more time is required:
   a. Clear the missing clock status by writing a 1 to MCDCR.MCLKCLR.
   b. Repeat steps 2 through 7. Do not reset the device. Doing so powers down the oscillator, which requires the procedure to be restarted from step 1.
   c. If the oscillator has not finished powering up in 10ms, there is a real clock failure.
7. If MCDCR.MCLKSTS is clear, the oscillator start up is a success. The system clock is now derived from XTAL.

3.7.8 Using an External Oscillator

The procedure for using an external oscillator connected to the X1 pin is similar to the procedure for using a crystal or resonator:

1. Clear the XTALCR.OSCOFF bit.
2. Set the XTALCR.SE bit to enable single-ended mode.
3. Clear the X1 counter by writing a 1 to X1CNT.CLR and keep clearing until the X1 counter value in the X1CNT register is no longer saturated 2047 (0x7FF).
4. Wait for the X1 counter value in the X1CNT register to reach 2047 (0x7FF).
5. Repeat steps 3 and 4 three additional times.
6. Select XTAL as the OSCCLK source by writing a 1 to CLKSRCCTL1.OSCCLKSRCSEL.
7. Check the MCLKSTS bit in the MCDCR register. If the bit set, either the external oscillator or the device has failed.
8. If MCLKSTS is clear, the switch to the external clock is a success. The system clock is now derived from XTAL.

3.7.9 Choosing PLL Settings

The equation shown in Figure 3-6 can be used to configure the PLL.

QDIV is the integer value of the multiplier.

PDIV is the reference divider for the OSCCLK.

RDIVCLK0 is the output divider of the PLLRAWCLK.

PLLSYSCLKDIV is the system clock divider.

For the permissible values of the multipliers and dividers, see the documentation for the respective registers.

Many combinations of multiplier and divider can produce the same output frequency. However, the product of the reference clock frequency and the multiplier (known as the VCO frequency) must be in the range specified in the *F28E12x Real-Time Microcontrollers Data Sheet*.

Note

The system clock frequency (PLLSYSCLK) can not exceed the limit specified in the *F28E12x Real-Time Microcontrollers Data Sheet*. This limit does not allow for oscillator tolerance.

3.7.10 System Clock Setup

Once the application requirements are understood, a specific clock configuration can be determined. The default configuration is for SYSOSCDIV4 to be used as the system clock (PLLSYSCLK) with a divider of 1. The following procedure can be used to set up the desired application configuration:

Refer to your device SysCtl_setClock() function inside C2000Ware installation for an example.

Recommended sequence to set up the system PLL:

1. Bypass the PLL by clearing SYSPLLCTL.PLLCLKEN. Allow at least 120 NOP instructions for this to take effect.
2. Power down the PLL by writing to SYSPLLCTL.PLLEN = 0. Allow at least 66 NOP instructions for this to take effect.
3. Select the reference clock source (OSCCLK) by writing to CLKSRCCTL1.OSCCLKSRCSEL. Allow at least 60 NOP instructions for this to take effect.
4. Set the system clock divider to /1 to make sure the fastest PLL configuration by clearing SYSCLKDIVSEL. PLLSYSCLKDIV.
5. Set PDIV, QDIV, and RDIVCLK0 simultaneously by writing 32-bit value in SYSPLLMULT at once. This automatically enables the PLL. Be sure the settings for multiplier and dividers do not violate the frequency specifications as defined in the F28E12x Real-Time Microcontrollers Data Sheet.
6. Wait for PLL to lock by polling for lock status bit to go high, SYSPLLSTS.LOCKS = 1.
7. Configure DCC with reference clock as OSCCLK and clock under measurement as PLLRAWCLK, and verify the frequency of the PLL. If the frequency is out of range, do not enable PLLRAWCLK as SYSCLK, stop here and troubleshoot. Refer to Chapter 7 for more information on the configuration and usage.
8. Switch to the PLL as the system clock by setting SYSPLLCTL.PLLCLKEN.

Note

1. SYSPLL must be bypassed and powered down manually before changing the OSCCLK source.
2. At least 120 CPU clock cycles delay is needed after bypassing PLL, that is, SYSPLLCTL.PLLCLKEN = 0.
3. At least 66 CPU clock cycles delay is needed after PLL is powered down, that is, SYSPLLCTL.PLLEN = 0.
4. At least 60 CPU clock cycles delay is needed after OSSCLK source is changed.
5. DCC can be used to check the validity of the PLL clock. This feature is included as part of SysCtl_setClock() function inside C2000Ware.
6. The PLL lock sequence should be retried in succession until the PLL is in the locked state and validated by the DCC to be working at the intended frequency.

3.7.11 SYS PLL Bypass

If the application requires the PLL clock to be bypassed from the system, then the application needs to configure SYSPLLCTL.PLLCLKEN = 0. It takes up to 60 CPU clock cycles before the bypass is effective. In the meantime if PLLSYSCLKDIV is reduced to a lower value (for example from /2 to /1, /4 to /2, and so on), the device can be clocked above the maximum rated frequency and can lead to unpredictable device behavior. Hence, a delay of 60 CPU clock cycles is required after bypassing the PLL from the enable state, that is, going from PLLCLKEN = 1 to PLLCLKEN = 0.

3.7.12 Clock (OSCCLK) Failure Detection

To achieve safety diagnostic, Missing Clock Detection (MCD) can be used.

Table 3-9 lists the details.

Table 3-9. Clock Source (OSCCLK) Failure Detection

Clock Failure Detection Circuitry	Clocks Detected	Time for Detection (in Cycles)	Limitations
Missing Clock Detection (MCD)	SYSOSC, XTAL/X1	8192 SECCLK cycles	Cannot detect WROSC clock failure.

3.7.12.1 Missing Clock Detection

The missing clock detect (MCD) logic detects OSCCLK failure, using SECCLK as the reference clock source. This circuit only detects complete loss of OSCCLK and doesn't do any detection of frequency drift on the OSCCLK.

This circuit monitors the OSCCLK (primary clock) using the clock provided by the SECCLK (secondary clock) as a backup clock. This circuit functions as below:

1. The primary clock (OSCCLK) clock keeps ticking a 7-bit counter (named as MCDPCNT). This counter is asynchronously reset with XRSn.
2. The secondary clock (SECCLK) clock keeps ticking a 13-bit counter (named as MCDSCNT). This counter is asynchronously reset with XRSn.
3. Each time MCDPCNT overflows, the MCDSCNT counter is reset. Thus, if OSCCLK is present or not slower than SECCLK by a factor of 64, MCDSCNT never overflows.
4. If OSCCLK stops for some reason or is slower than SECCLK by at least a factor of 64, the MCDSCNT overflows and a missing clock condition is detected on OSCCLK.
5. The above check is continuously active, unless the MCD is disabled using MCDCR register (by making the MCLKOFF bit 1).
6. If the circuit ever detects a missing OSCCLK, the following occurs:
   - The MCDSTS flag is set.
   - The MCDSCNT counter is frozen to prevent further missing clock detection.
   - The CLOCKFAIL signal goes high, which generates TRIP events to PWM modules and sends NMIs to CPU1.NMIWD.
   - PLL is forcefully bypassed and OSCCLK source is switched to SECCLK (New, System Clock Frequency = SECCLK Freq 10MHz)/SYSDIV). In the meantime when the clock switches to SECCLK, the System runs on PLL limp Clock.
   - SYSPLLMULT.IMULT is zeroed out automatically in this case.
   - While the MCDSTS bit is set, the OSCCLKSRCSEL bits have no effect and OSCCLK is forcefully connected to SECCLK.
   - PLLRAWCLK going to the system is switched to SECCLK automatically.
7. If the MCLKCLR bit is written (this is a W=1 bit), MCDSTS bit is cleared and OSCCLK source is decided by the OSCCLKSRCSEL bits. Writing to MCLKCLR also clears the MCDPCNT and MCDSCNT counters to allow the circuit re-evaluate missing clock detection. If user wants to lock the PLL after missing clock detection, switch the clock source to SECCLK (using OSCCLKSRCSEL register), do a MCLKCLR and re-lock the PLL.
8. The MCD is enabled at power up.

Figure 3-10 shows the missing clock logic functional flow.

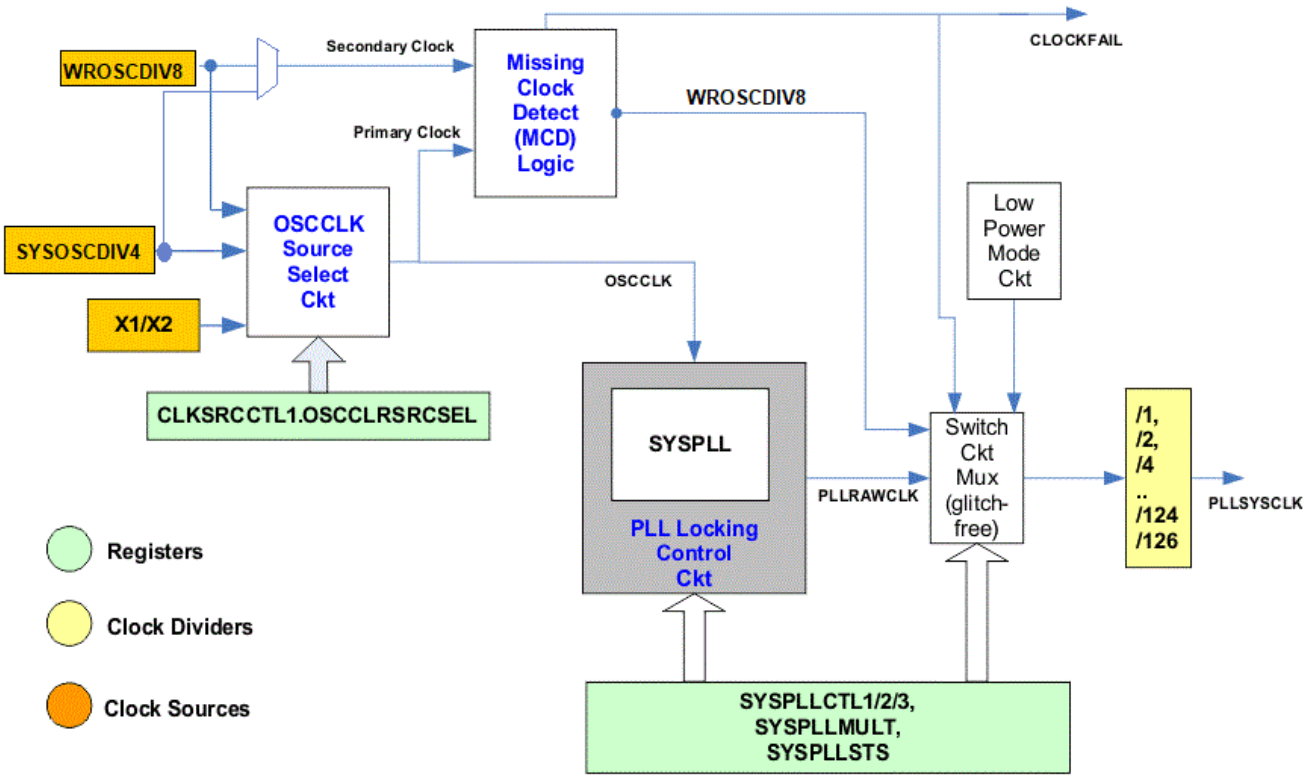


Figure 3-10. Missing Clock Detection Logic

Note

1. When XTAL is primary clock, SYSOSCDIV4 can be selected as secondary clock as this clock is more accurate than WROSCDIV8.
2. On a complete clock failure when OSCCLK is not operating, the operation can take a maximum time of 8192 SECCLK cycles (that is, 0.8192ms) before the CLOCKFAIL signal goes high, after which:

- NMI is generated
- OSCCLK is switched to SECCLK
- PWM Trip happens

3.8 32-Bit CPU Timers 0/1/2

This section describes the three 32-bit CPU timers (TIMER0/1/2) shown in Figure 3-11. Timer0 and Timer1 can be used in user applications. Timer2 is reserved for real-time operating system uses (for example, TI-RTOS). If the application is not using an operating system that utilizes this timer, then Timer2 can be used in the application. Timer interrupt signals (TINT0, TINT1, TINT2) are connected as shown in Figure 3-12.

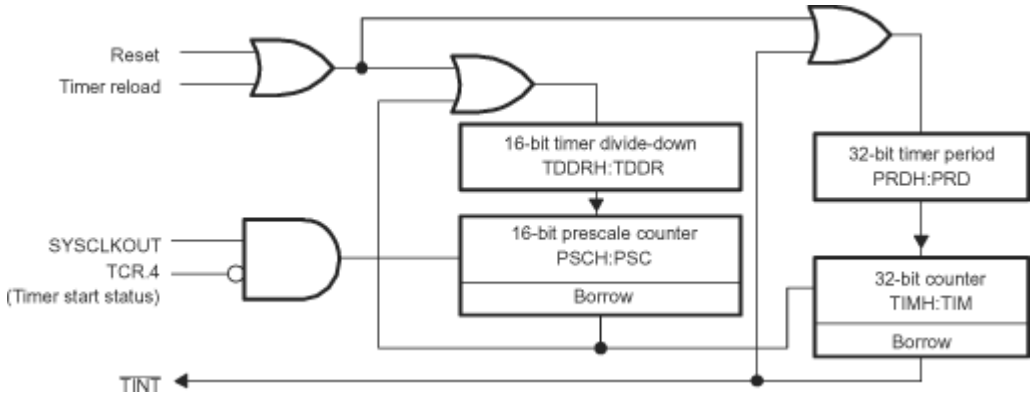
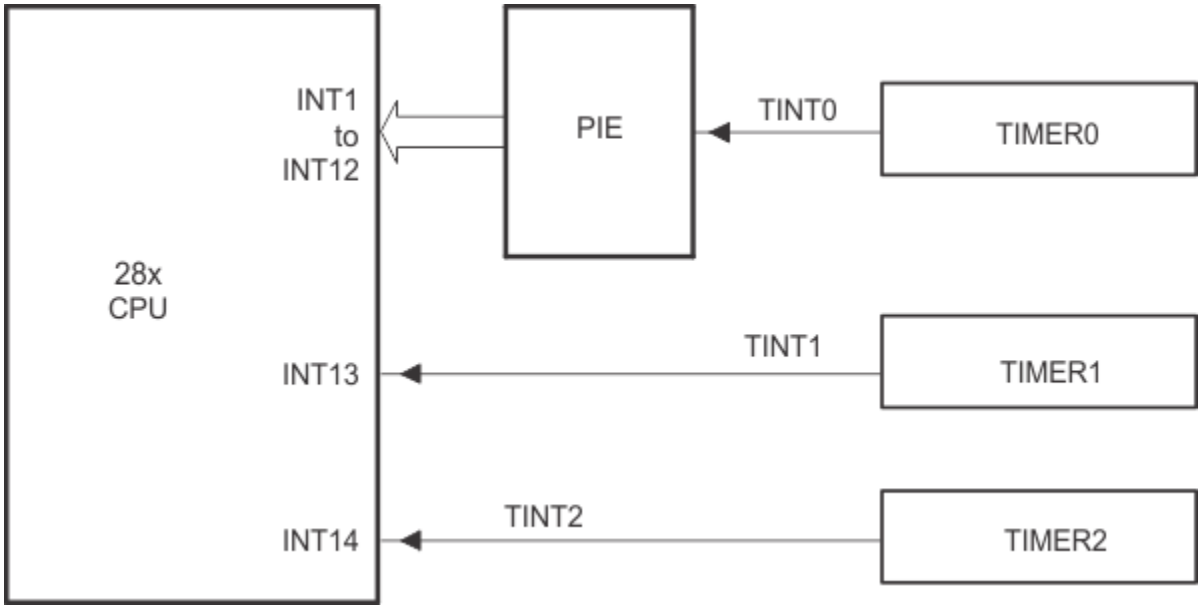


Figure 3-11. CPU Timers



A. The timer registers are connected to the memory bus of the C28x processor.
B. The CPU timers are synchronized to SYSCLKOUT.

Figure 3-12. CPU Timer Interrupt Signals and Output Signal

The general operation of a CPU timer is as follows:

- The 32-bit counter register, TIMH:TIM, is loaded with the value in the period register PRDH:PRD.
- The counter decrements once every (TPR[TDDRH:TDDR] + 1) SYSCLK cycles, where TDDRH:TDDR is the timer divider.
- When the counter reaches 0, a timer interrupt output signal generates an interrupt pulse.

The registers listed in Section 3.15 are used to configure the timers.

3.9 Watchdog Timer

The watchdog module consists of an 8-bit counter sourced by a prescaled clock (WDCLK, which is connected to WROSCDIV8). When the counter reaches the maximum value, the module generates an output pulse 512 WDCLKs wide. This pulse can generate an interrupt or a reset. The CPU must periodically write a 0x55 + 0xAA sequence into the watchdog key register to reset the watchdog counter. The counter can also be disabled.

The counter's clock is divided down from WDCLK by two dividers. The prescaler is adjustable from /1 to /64 in powers of two. The pre-divider defaults to /512 for backwards compatibility, but is adjustable from /2 to /4096 in powers of two. This allows a wide range of timeout values for safety-critical applications.

Figure 3-13 shows the various functional blocks within the watchdog module.

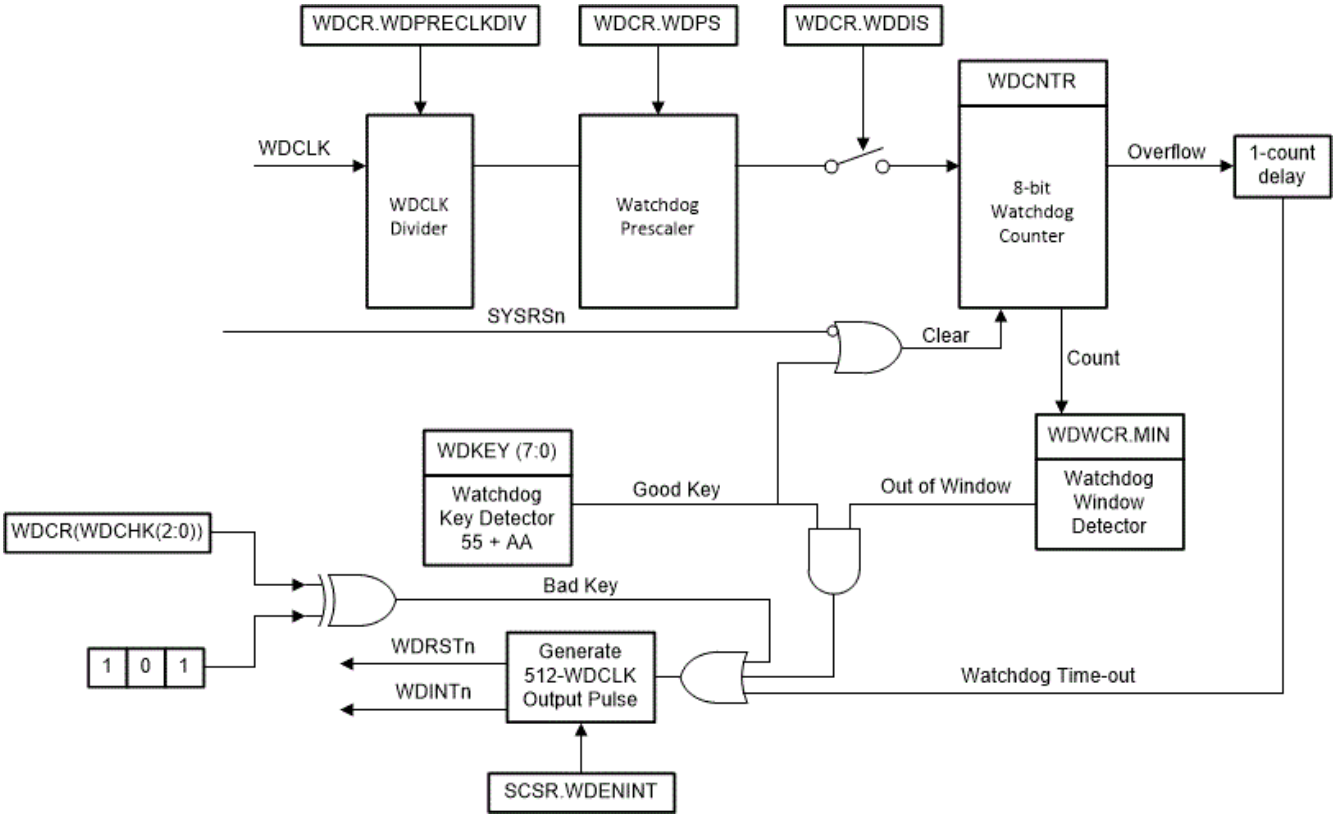


Figure 3-13. Watchdog Timer Module

3.9.1 Servicing the Watchdog Timer

The watchdog counter (WDCNTR) is reset when the proper sequence is written to the WDKEY register before the 8-bit watchdog counter overflows. The WDCNTR is reset-enabled when a value of 0x55 is written to the WDKEY. When the next value written to the WDKEY register is 0xAA, then the WDCNTR is reset. Any value written to the WDKEY other than 0x55 or 0xAA causes no action. Any sequence of 0x55 and 0xAA values can be written to the WDKEY without causing a system reset; only a write of 0x55 followed by a write of 0xAA to the WDKEY resets the WDCNTR.

The first action that enables the WDCNTR to be reset is shown in Step 3 in Table 3-10. The WDCNTR is not actually reset until step 6. Step 8 again re-enables the WDCNTR to be reset and step 9 resets the WDCNTR. Step 10 again re-enables the WDCNTR to be reset. Writing the wrong key value to the WDKEY in step 11 causes no action, however the WDCNTR is no longer enabled to be reset and the 0xAA in step 12 now has no effect.

If the watchdog is configured to reset the device, then a WDCR overflow or writing the incorrect value to the WDCR[WDCHK] bits resets the device and set the watchdog flag (WDRSn) in the reset cause register (RESC). After a reset, the program can read the state of this flag to determine whether the reset was caused by the watchdog. After doing this, the program can clear WDRSn to allow subsequent watchdog resets to be detected. Watchdog resets are not prevented when the flag is set.

Table 3-10. Example Watchdog Key Sequences

Step	Value Written to WDKEY	Result
1	0xAA	No action
2	0xAA	No action
3	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
4	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
5	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
6	0xAA	WDCNTR is reset.
7	0xAA	No action
8	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
9	0xAA	WDCNTR is reset.
10	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
11	0x32	Improper value written to WDKEY. No action, WDCNTR no longer enabled to be reset by next 0xAA.
12	0xAA	No action due to previous invalid value.
13	0x55	WDCNTR is enabled to be reset if next value is 0xAA.
14	0xAA	WDCNTR is reset.

3.9.2 Minimum Window Check

To complement the timeout mechanism, the watchdog also contains an optional "windowing" feature that requires a minimum delay between counter resets. This can help protect against error conditions that bypass large parts of the normal program flow but still include watchdog handling. Since watchdog runs on WROSCDIV8 (2.5MHz to 8.75MHz range), the WDWCR.MIN value configuration and the WDKEY update periodicity by software have to consider the frequency variance of WROSCDIV8. Table 3-11 shows the minimum and maximum values of window.

To set the window minimum, write the desired minimum watchdog count to the WDWCR register. This value takes effect after the next WDKEY sequence. From then on, any attempt to service the watchdog when WDCNTR is less than WDWCR triggers a watchdog interrupt or reset. When WDCNTR is greater than or equal to WDWCR, the watchdog can be serviced normally.

At reset, the window minimum is zero, which disables the windowing feature.

Table 3-11. Minimum Window Configuration

WDCLK PRESCALE	WDCLK PRECLKDIV	WDWCR.MIN	WROSCDIV8 Frequency (Hz)	WDCLK Period (sec)	Window Minimum (sec)	Window Maximum (sec)
1	512	32	2.50E+06	2.05E-04	2.05E-04	5.24E-02
			8.75E+06	5.85E-05	1.87E-03	1.50E-02

3.9.3 Watchdog Reset or Watchdog Interrupt Mode

The watchdog can be configured in the SCSR register to either reset the device ($\overline{\text{WDRST}}$) or assert an interrupt ($\overline{\text{WDINT}}$), if the watchdog counter reaches the maximum value. The behavior of each condition is:

- **Reset mode:** If the watchdog is configured to reset the device, then the $\overline{\text{WDRST}}$ signal pulls the device reset ($\overline{\text{XRS}}$) pin low for 512 SECCLK cycles when the watchdog counter reaches the maximum value.
- **Interrupt mode:** When the watchdog counter expires, an interrupt is asserted by driving the $\overline{\text{WDINT}}$ signal low for 512 SECCLK cycles. The falling edge of $\overline{\text{WDINT}}$ triggers a WAKEINT interrupt in the PIE if enabled. Because the PIE is edge-triggered, re-enabling the WAKEINT while $\overline{\text{WDINT}}$ is active does not produce a duplicate interrupt.

  To avoid unexpected behavior, software must not change the configuration of the watchdog while $\overline{\text{WDINT}}$ is active. For example, changing from interrupt mode to reset mode while $\overline{\text{WDINT}}$ is active immediately resets the device. Disabling the watchdog while $\overline{\text{WDINT}}$ is active causes a duplicate interrupt if the watchdog is later re-enabled. If a debug reset is issued while $\overline{\text{WDINT}}$ is active, the reset cause register (RESC) shows a watchdog reset. The WDINTS bit in the SCSR register can be read to determine the current state of $\overline{\text{WDINT}}$.

3.9.4 Watchdog Operation in Low Power-Modes

In IDLE mode, the watchdog interrupt ($\overline{\text{WDINT}}$) signal can generate an interrupt to the CPU to take the CPU out of IDLE mode. As with any other peripheral, the watchdog interrupt triggers a WAKE interrupt in the PIE during IDLE mode. Software must determine which peripheral caused the interrupt.

Note

If the watchdog interrupt is used to wake-up from an IDLE low-power mode condition, software must make sure that the $\overline{\text{WDINT}}$ signal goes back high before attempting to reenter the IDLE mode. The $\overline{\text{WDINT}}$ signal is held low for 512 SECCLK cycles when the watchdog interrupt is generated. The current state of $\overline{\text{WDINT}}$ can be determined by reading the watchdog interrupt status bit (WDINTS) bit in the SCSR register. WDINTS follows the state of $\overline{\text{WDINT}}$ by 2 SYSCLKOUT cycles.

In HALT mode, the internal oscillators and watchdog timer are kept active if the user sets CLKSRCCTL1.WDHALTI = 1. A watchdog reset can wake the system from HALT mode, but a watchdog interrupt cannot wake the system.

3.9.5 Emulation Considerations

The watchdog module behaves as follows under various debug conditions:

CPU Suspended	When the CPU is suspended, the watchdog clock (WDCLK) is suspended.
Run-Free Mode	When the CPU is placed in run-free mode, then the watchdog module resumes operation as normal.
Real-Time Single-Step Mode	When the CPU is in real-time single-step mode, the watchdog clock (WDCLK) is suspended. The watchdog remains suspended even within real-time interrupts.
Real-Time Run-Free Mode	When the CPU is in real-time run-free mode, the watchdog operates as normal.

3.10 Low-Power Modes

This device has HALT, IDLE, and STANDBY as clock-gating low-power modes.

All low-power modes are entered by setting the LPMCR register and executing the IDLE instruction. More information about this instruction can be found in the *TMS320C28x CPU and Instruction Set Reference Guide*.

Low-power modes must not be entered into while a Flash program or erase operation is ongoing. Entering HALT stops all CPU and peripheral activities. This includes active transmissions and control algorithms. When preparing to enter HALT mode, the application must make sure that the system is prepared to enter a period of inactivity.

Before entering HALT mode check the value of the GPIODAT register of the pin selected for HALT wake-up (GPIOLPMSEL0/1) prior to entering the low-power mode to make sure that the wake event has not already been asserted.

3.10.1 Clock-Gating Low-Power Modes

IDLE and HALT modes on this device are similar to those on other C28x devices. Table 3-12 describes the effect on the system when any of the clock-gating low-power modes are entered.

Table 3-12. Effect of Clock-Gating Low-Power Modes on the Device

Modules/ Clock Domain	IDLE	STANDBY	HALT
SYSCLK	Active	Gated	Gated
CPUCLK	Gated	Gated	Gated
Clock to modules connected to PERx.SYSCLK	Active	Gated	Gated
WDCLK	Active	Active	Gated if CLKSRCCTL1.WDHALTI = 0
PLL	Powered	Powered	Software must power down PLL before entering HALT.
Flash(1)	Powered	Powered	Powered
XTAL(2)	Powered	Powered	Powered

(1) The Flash module is not powered down by hardware in any LPM. The Flash module can be powered down using software, if required by the application. For more information, see Chapter 6.
(2) The XTAL is not powered down by hardware in any LPM. The XTAL can be powered down using software by setting the XTALCR.OSCOFF bit to 1. This can be done at any time during the application, if the XTAL is not required.

Note

Writing the WDHALTI bit unlocks the PLL and clear the SYSPLLSTS.LOCKS bit.

3.10.2 IDLE

IDLE is a standard feature of the C28x CPU. In this mode, the CPU clock is gated while all peripheral clocks are left running. IDLE can thus be used to conserve power while a CPU is waiting for peripheral events.

Any enabled interrupt wakes up the CPU from IDLE mode.

To enter IDLE mode, set LPMCR.LPM to 0x0 and execute the IDLE instruction.

The CPU resumes normal operations upon any enabled interrupt event.

3.10.3 STANDBY

STANDBY is a more aggressive low-power mode that gates both the CPU clock and any peripheral clocks derived from the CPU's SYSCLK. The watchdog, however, is left active. STANDBY is best designed for an application where the wake-up signal comes from an external system (or CPU subsystem) rather than a peripheral input.

An NMI (or optionally, a watchdog interrupt or a configured GPIO) can wake the CPU from STANDBY mode. Each GPIO can be configured to wake the CPU when the GPIOs are driven active-low. Upon wakeup, the CPU receives the WAKEINT interrupt if configured.

To enter STANDBY mode:

1. Set LPMCR.LPM to 0x1.
2. Enable the WAKEINT interrupt in the PIE.
3. For watchdog interrupt wakeup, set LPMCR.WDINTE to 1 and configure the watchdog to generate interrupts.
4. For GPIO wakeup, set GPIOLPMSEL0 and GPIOLPMSEL1 to connect the chosen GPIOs to the LPM module, and set LPMCR.QUALSTDBY to select the number of OSCCLK cycles for input qualification.
5. Execute the IDLE instruction to enter STANDBY.

To wake up from STANDBY mode:

1. Configure the desired GPIO to trigger the wakeup.
2. Drive the selected GPIO signal low; the signal must remain low for the number of OSCCLK cycles specified in the QUALSTDBY bits in the LPMCR register. If the signal is sampled high during this period, the count restarts.

At the end of the qualification period, the PLL enables the CLKIN to the CPU and the WAKEINT interrupt is latched in the PIE block.

The CPU is now out of STANDBY mode and can resume normal execution.

3.10.4 HALT

HALT is a global low-power mode that gates almost all system clocks and allows for power-down of oscillators and analog blocks.

Unlike on other C2000™ devices, HALT mode does not automatically power down the XTAL upon HALT entry. Additionally, if the XTAL is not powered on, waking up from HALT mode does not automatically power on the XTAL. The XTALCR.OSCOFF bit has been added to power on and off the XTAL circuitry when not needed through application software.

For applications that require minimal power consumption during HALT mode, application software must power off the XTAL prior to entering HALT. If the OSCCLK source is configured to be XTAL, the application must first switch the OSSCLK source to WROSCDIV8 or SYSOSCDIV4 prior to setting XTALCR.OSCOFF.

Each GPIO can be configured to wake up the system from HALT. No other wakeup option is available. However, the watchdog timer can still be clocked and can be configured to produce a watchdog reset if a timeout mechanism is needed. On wakeup, the CPU receives a WAKEINT interrupt.

To enter HALT mode:

1. Enable the WAKEINT interrupt in the PIE.
2. Set LPMCR.LPM to 0x2. Set GPIOLPMSEL0 and GPIOLPMSEL1 to connect the chosen GPIOs to the LPM module.
3. Set CLKSRCCTL1.WDHALTI to 1 to keep the watchdog timer active and WROSCDIV8 and SYSOSCDIV4 powered up in HALT.
4. Set CLKSRCCTL1.WDHALTI to 0 to disable the watchdog timer.
5. Execute the IDLE instruction to enter HALT.

If an interrupt or NMI is received while the IDLE instruction is in the pipeline, the system begins executing the WAKEINT ISR. After HALT wakeup, ISR execution resumes where execution left off.

Note

Before entering HALT mode, if the system PLL is locked (SYSPLL.LOCKS = 1), the PLL must also be connected to the system clock (PLLCTL.PLLCLKEN = 1). Otherwise, the device never wakes up.

To wake up from HALT mode:

1. Drive the selected GPIO low for a minimum 5µs. This activates the WAKEINT PIE interrupt.
2. Drive the wake-up GPIO high again to initiate the powering up of the SYSPLL
3. Wait 16µs plus 1024 OSCLK cycles to allow the PLLs to lock and the WAKEINT ISR to be latched.
4. Execute the WAKEINT ISR.

The device is now out of HALT mode and can resume normal execution.

3.11 Memory Controller Module

On this device, the M0, M1, and GSx RAMs are dedicated to the CPU.

All these RAMs are highly configurable to achieve control for write access and fetch access from different controllers. All dedicated RAMs are enabled with the ECC feature (both data and address) and shared RAMs are enabled with the parity feature (both data and address). Some of the dedicated memories are secure memory as well. Refer to Chapter 5 for more details. Each RAM has a controller that takes care of the access protection/ security related checks and ECC/Parity features for that RAM. Figure 3-14 shows the configuration of these RAMs.

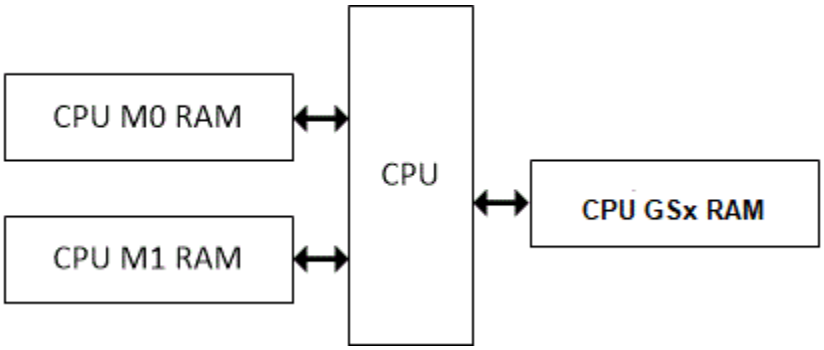


Figure 3-14. Memory Architecture

3.11.1 Dedicated RAM (Mx RAM)

This device has two dedicated RAM blocks: M0 and M1. M0 and M1 memories are small blocks of memory that are tightly coupled with the CPU. Only the CPU has access to these memories. No other controllers have access to these memories.

All dedicated RAMs have the ECC feature.

3.11.2 Global Shared RAM (GSx RAM)

RAM blocks that are accessible from the CPU and DMA are called global shared RAMs (GSx RAMs). Table 3-13 shows the features of the GSx RAM.

The shared RAM has different levels of access protection which can be enabled or disabled by configuring specific bits in the GSxACCPROT registers.

Access protection configuration for the GSx RAM block can be locked by the user to prevent further updates to this bit field. The user can also choose to permanently lock the configuration to individual bit fields by setting the specific bit fields in the GSxCOMMIT register (refer to the register description for more details). Once a configuration is committed for a particular GSx RAM block, the configuration can not be changed further until a CPU.$\overline{\text{SYSRS}}$ is issued.

Table 3-13. Global Shared RAM

CPU (Fetch)	CPU (Read)	CPU (Write)	CPU.DMA (Read)	CPU.DMA (Write)
Yes	Yes	Yes	Yes	Yes

3.11.3 Access Arbitration

For a shared RAM, multiple accesses can happen at any given time. The maximum number of accesses to any shared RAM at any given time depends on the type of shared RAM. On this device, a combination of a fixed and round robin scheme is followed to arbitrate multiple access at any given time.

The following is the order of fixed priority for CPU accesses:

1. Data Write/Program Write
2. Data Read
3. Program Read/Program Fetch

Figure 3-15 represents the arbitration scheme on global shared memories.

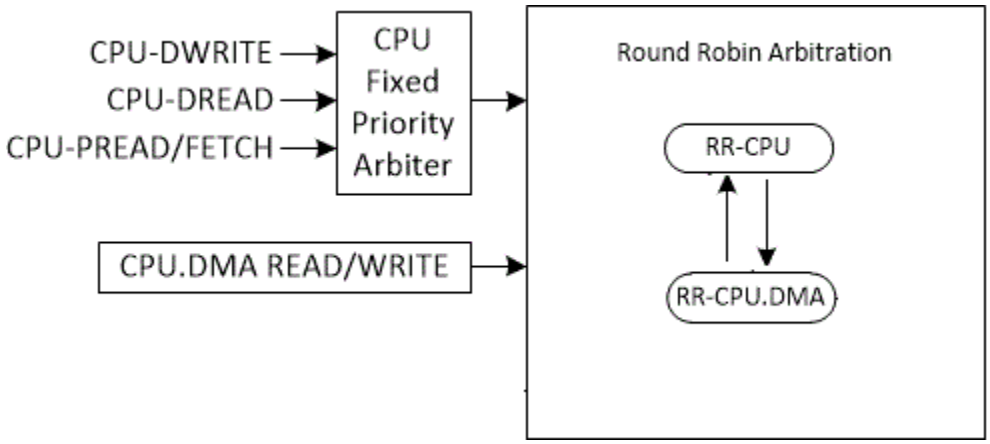


Figure 3-15. Arbitration Scheme on Global Shared Memories

3.11.4 Memory Error Detection, Correction, and Error Handling

These devices have memory error detection and correction features to satisfy safety standards requirements. These requirements warrant the addition of detection mechanisms for finite dangerous failures.

In this device, all the shared RAMs have parity protection. The parity scheme used is even parity. Parity covers the data bits stored in memory as well as address.

Parity calculation is done inside the memory controller module and calculated. Parity is written into the memory along with the data. Parity is computed for 16-bit data; hence, for each 32-bits of data, there are three 3-bit parity, two of the codes are for data and a third code for the address.

3.11.4.1 Error Detection and Correction

Error detection is done while reading the data from memory. The error detection is performed for data as well as address. For parity memory, only a single-bit error gets detected. These errors are called uncorrectable errors.

3.11.4.2 Error Handling

For each correctable error, the count in the correctable error count register increments by one. When the value in this count register becomes equal to the value configured in the correctable error threshold register, an interrupt is generated to the CPU, if the interrupt is enabled in the correctable interrupt enable register. The user needs to configure the correctable error threshold register based on the system requirements. Also, the address for which the error occurred, gets latched into a register and a flag also gets set in a status register.

If there are uncorrectable errors, an NMI gets generated for the CPU. In this case also, the address for which the error occurred gets latched into a register, and a flag gets set in a status register.

Table 3-14 summarizes different error situations that can arise. These need to be handled appropriately in the software, using the status and interrupt indications provided.

Table 3-14. Error Handling in Different Scenarios

Access Type	Error Found In	Error Type	Status Indication	Error Notification
Reads	Data read from memory	Uncorrectable Error (Single-bit error for Parity RAMs or Double bit Error for ECC RAMs)	Yes - CPU Read Error Address Register Data returned to CPU is incorrect	NMI for CPU access
Reads	Data read from memory	Single-bit error for ECC RAMs	Yes - CPU Read Error Address Register Increment single error counter	Interrupt when error counter reaches the user programmable threshold for single errors
Reads	Data read from PIE memory	Parity error	Yes - PIE Parity Error sets bit in MEM_CFG_REGS	Bit set in MEM_CFG_REGS
Reads	Address	Address error	Yes - CPU Read Address Error Register Data returned to CPU is incorrect	NMI to CPU for CPU access

Note

In the case of an uncorrectable error during fetch on the CPU, there is the possibility of getting an ITRAP before an NMI exception, since garbage instructions enter into the CPU pipeline before the NMI gets generated.

During debug accesses, correctable and uncorrectable errors are masked.

3.11.5 Application Test Hooks for Error Detection and Correction

Since error detection and correction logic is part of safety critical logic, safety applications need to make sure that the logic is always working fine (during run time also). To enable this, a test mode is provided, in which a user can modify the data bits (without modifying the Parity bits) or Parity bits directly. Using this feature, an Parity error can be injected into data.

Note

The memory-map for Parity bits and data bits are the same. The user must choose a different test mode to access Parity bits. In test mode, all access to memories (data as well as Parity) must be done as 32-bit access only.

Table 3-15 shows the bit mapping for the Parity bits when the bits are read in RAMTEST mode using the respective addresses.

Table 3-15. Mapping of Parity Bits in Read Data from Parity Address Map

Data Bits Location in Read Data	Content (Parity Memory)
0	Parity for lower 16 bits of data
7:1	Not Used
8	Parity for upper 16 bits of data
15:9	Not Used
16	Parity for address
31:17	Not Used

3.11.6 RAM Initialization

To make sure that read/fetch from uninitialized RAM locations do not cause parity errors, the RAM_INIT feature is provided for each memory block. Using this feature, any RAM block can be initialized with 0x0 data and respective Parity bits accordingly. This can be initiated by setting the INIT bit to 1 for the specific RAM block in INIT registers. To check the status of RAM initialization, software can poll for the INITDONE bit for that RAM block in the INITDONE register to be set. Unless this bit gets set, no access must be made to that RAM memory block.

Note

None of the masters can access the memory while initialization is taking place. If memory is accessed before RAMINITDONE is set, the memory read/write as well as initialization does not happen correctly.

3.12 JTAG

Gel files perform certain initialization tasks. This helps the users in a debug environment. However, when executed standalone (without the emulator connected) the application can not work as expected, since there is no gel file to perform those initializations. For example, gel file disables watchdog. If user code does not service the watchdog in the application (or fails to disable the watchdog), there is a difference in how the application behaves with the debugger and without the debugger.

Common tasks performed by the gel files (but not boot-ROM):

- On Reset:
  - Disable Flash ECC on some devices.
    - Disabling ECC only when using Flash API functions, see the Flash API User Guide for details. Otherwise, TI suggests to always program ECC and enable ECC-check.
  - Disable Watchdog
  - Select real-time mode or C28x mode
- On Restart:
  - Select real-time mode or C28x mode
  - Clear IER and IFR
- On Target Connect:
  - Select real-time mode or C28x mode

For more information, see *C2000 MCU JTAG Connectivity Debug Application Note*.

3.12.1 JTAG Noise and TAP_STATUS

The TAP_STATUS register reflects the status of the JTAG TAP at any given time. Normally when no JTAG is connected to the device, the status can be IDLE. In some cases with excessive PCB noise, there can be unwanted TMS and TCK toggles that take JTAG out of the IDLE state. When persistent, this can ultimately lead to unwanted activation of the JTAG Boundary Scan or some other JTAG mode that can interfere with the intended application. To avoid this scenario, place strong enough pull resistors on the board to prevent noise from activating JTAG. As a debug tool, the TAP_STATUS register can be polled by the application code to detect if this is a cause of device disturbance. The SOFTPRES40[JTAG_nTRST] register can also be used to reset the JTAG TAP through software. Use this reset register with caution, as this prevents connecting a debugger unless the code qualifies writes to this register with some other GPIO state or other means to distinguish between noise and debugger accesses.

3.13 System Control Register Configuration Restrictions

Memory-mapped registers in the System Control operate on OSCCLK clock domain; hence, any CPU writes to these registers requires a delay between subsequent writes otherwise a second write can be lost. The application needs to take this into consideration and add a delay in terms of the number of NOP instructions after every write to these registers that are mentioned in Table 3-16. The formula to compute delay between subsequent writes:

Delay (in SYSCLK cycles) = 3 × (F_{SYSCLK} ÷ F_{OSCCLK}) + 9

For Example - For SYSCLK = 160MHz

Delay (in SYSCLK cycles) = 3 × (160MHz ÷ 8MHz) + 9 = 69 SYSCLK cycles

Table 3-16. System Control Registers Impacted

Registers requiring delay after every write
PERCLKDIVSEL
SYSCLKDIVSEL
SYSPLLCTL
SYSPLLMULT
WDCR
XCLKOUTDIVSEL
XTALCR
CLKSRCCTL1
CLKSRCCTL2
CLKSRCCTL3
CPU1TMR2CTL (TMR2CLKCTL)

3.14 Software

3.14.1 SYSCTL Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/sysctl

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

3.14.1.1 Missing clock detection (MCD)

FILE: sysctl_ex1_missing_clock_detection.c

This example demonstrates the missing clock detection functionality and the way to handle it. Once the MCD is simulated by disconnecting the OSCCLK to the MCD module an NMI would be generated. This NMI determines that an MCD was generated due to a clock failure which is handled in the ISR.

Before an MCD the clock frequency would be as per device initialization (160Mhz). Post MCD the frequency would move to SYSOSCDIV4.

The example also shows how we can lock the PLL after missing clock, detection, by first explicitly switching the clock source to SYSOSCDIV4, resetting the missing clock detect circuit and then re-locking the PLL. Post a re-lock the clock frequency would be 160Mhz but using the SYSOSCDIV4 as clock source.

External Connections

- None.

Watch Variables

- *fail* - Indicates that a missing clock was either not detected or was not handled correctly.
- *mcd_clkfail_isr* - Indicates that the missing clock failure caused an NMI to be triggered and called an the ISR to handle it.
- *mcd_detect* - Indicates that a missing clock was detected.
- *result* - Status of a successful handling of missing clock detection

3.14.1.2 XCLKOUT (External Clock Output) Configuration

FILE: sysctl_ex2_xclkout_config.c

This example demonstrates how to configure the XCLKOUT pin for observing internal clocks through an external pin, for debugging and testing purposes.

In this example, we are using SYSOSCDIV4 as the XCLKOUT clock source and configuring the divider as 8. Expected frequency of XCLKOUT = (SYSOSCDIV4 freq)/8 = 8/8 = 1MHz

View the XCLKOUT on GPIO16 using an oscilloscope.

3.15 SYSCTRL Registers

This section describes the SYSCTRL registers.

3.15.1 SYSCTRL Base Address Table

Table 3-17. SYSCTRL Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
CpuTimer0Regs	CPUTIMER_REGS	CPUTIMER0_BASE	0x0000_0C00	-	-
CpuTimer1Regs	CPUTIMER_REGS	CPUTIMER1_BASE	0x0000_0C08	-	-
CpuTimer2Regs	CPUTIMER_REGS	CPUTIMER2_BASE	0x0000_0C10	-	-
PieCtrlRegs	PIE_CTRL_REGS	PIECTRL_BASE	0x0000_0CE0	-	-
PieVectTable	PIE_VECT_TABLE	PIEVECTTABLE_BASE	0x0000_0D00	-	-
WdRegs	WD_REGS	WD_BASE	0x0000_7000	-	YES
NmiIntruptRegs	NMI_INTRUPT_REGS	NMI_BASE	0x0000_7060	-	YES
XintRegs	XINT_REGS	XINT_BASE	0x0000_7070	-	YES
SyncSocRegs	SYNC_SOC_REGS	SYNCSOC_BASE	0x0000_7940	-	YES
DmaClaSrcSelRegs	DMA_CLA_SRC_SEL_REGS	DMACLASRCSEL_BASE	0x0000_7980	-	YES
DevCfgRegs	DEV_CFG_REGS	DEVCFG_BASE	0x0005_D000	-	YES
ClkCfgRegs	CLK_CFG_REGS	CLKCFG_BASE	0x0005_D200	-	YES
CpuSysRegs	CPU_SYS_REGS	CPUSYS_BASE	0x0005_D300	-	YES
SysStatusRegs	SYS_STATUS_REGS	SYSSTAT_BASE	0x0005_D400	-	YES
MemCfgRegs	MEM_CFG_REGS	MEMCFG_BASE	0x0005_F400	-	YES
MemoryErrorRegs	MEMORY_ERROR_REGS	MEMORYERROR_BASE	0x0005_F540	-	YES
RomWaitStateRegs	ROM_WAIT_STATE_REGS	ROMWAITSTATE_BASE	0x0005_F580	-	YES
TestErrorRegs	TEST_ERROR_REGS	TESTERROR_BASE	0x0005_F590	-	YES
UidRegs	UID_REGS	UID_BASE	0x0007_2172	-	-

3.15.2 CPUTIMER_REGS Registers

Table 3-18 lists the memory-mapped registers for the CPUTIMER_REGS registers. All register offset addresses not listed in Table 3-18 should be considered as reserved locations and the register contents should not be modified.

Table 3-18. CPUTIMER_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	TIM	CPU-Timer, Counter Register	
2h	PRD	CPU-Timer, Period Register	
4h	TCR	CPU-Timer, Control Register	
6h	TPR	CPU-Timer, Prescale Register	
7h	TPRH	CPU-Timer, Prescale Register High	

Complex bit access types are encoded to fit into small table cells. Table 3-19 shows the codes that are used for access types in this section.

Table 3-19. CPUTIMER_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
Reset or Default Value		
-n		Value after reset or the default value

3.15.2.1 TIM Register (Offset = 0h) [Reset = 0000FFFFh]

TIM is shown in Figure 3-16 and described in Table 3-20.

Return to the Summary Table.

CPU-Timer, Counter Register

Figure 3-16. TIM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSW																LSW															
R/W-0h																R/W-FFFFh															

Table 3-20. TIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MSW	R/W	0h	CPU-Timer Counter Registers The TIMH register holds the high 16 bits of the current 32-bit count of the timer. The TIMH:TIM decrements by one every (TDDRH:TDDR+1) clock cycles, where TDDRH:TDDR is the timer prescale dividedown value. When the TIMH:TIM decrements to zero, the TIMH:TIM register is reloaded with the period value contained in the PRDH:PRD registers. The timer interrupt (TINT) signal is generated. Reset type: SYSRSn
15-0	LSW	R/W	FFFFh	CPU-Timer Counter Registers The TIM register holds the low 16 bits of the current 32-bit count of the timer. The TIMH:TIM decrements by one every (TDDRH:TDDR+1) clock cycles, where TDDRH:TDDR is the timer prescale dividedown value. When the TIMH:TIM decrements to zero, the TIMH:TIM register is reloaded with the period value contained in the PRDH:PRD registers. The timer interrupt (TINT) signal is generated. Reset type: SYSRSn

3.15.2.2 PRD Register (Offset = 2h) [Reset = 0000FFFFh]

PRD is shown in Figure 3-17 and described in Table 3-21.

Return to the Summary Table.

CPU-Timer, Period Register

Figure 3-17. PRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSW																LSW															
R/W-0h																R/W-FFFFh															

Table 3-21. PRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MSW	R/W	0h	CPU-Timer Period Registers The PRDH register holds the high 16 bits of the 32-bit period. When the TIMH:TIM decrements to zero, the TIMH:TIM register is reloaded with the period value contained in the PRDH:PRD registers, at the start of the next timer input clock cycle (the output of the prescaler). The PRDH:PRD contents are also loaded into the TIMH:TIM when you set the timer reload bit (TRB) in the Timer Control Register (TCR). Reset type: SYSRSn
15-0	LSW	R/W	FFFFh	CPU-Timer Period Registers The PRD register holds the low 16 bits of the 32-bit period. When the TIMH:TIM decrements to zero, the TIMH:TIM register is reloaded with the period value contained in the PRDH:PRD registers, at the start of the next timer input clock cycle (the output of the prescaler). The PRDH:PRD contents are also loaded into the TIMH:TIM when you set the timer reload bit (TRB) in the Timer Control Register (TCR). Reset type: SYSRSn

3.15.2.3 TCR Register (Offset = 4h) [Reset = 0001h]

TCR is shown in Figure 3-18 and described in Table 3-22.

Return to the Summary Table.

CPU-Timer, Control Register

Figure 3-18. TCR Register

15	14	13	12	11	10	9	8
TIF	TIE	RESERVED		FREE	SOFT	RESERVED	
R/W1C-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	R-0h	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED		TRB	TSS	RESERVED			
R-0h		R/W-0h	R/W-0h	R-1h			

Table 3-22. TCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	TIF	R/W1C	0h	CPU-Timer Overflow Flag. TIF indicates whether a timer overflow has happened since TIF was last cleared. TIF is not cleared automatically and does not need to be cleared to enable the next timer interrupt. Reset type: SYSRSn 0h (R/W) = The CPU-Timer has not decremented to zero. Writes of 0 are ignored. 1h (R/W) = This flag gets set when the CPU-timer decrements to zero. Writing a 1 to this bit clears the flag.
14	TIE	R/W	0h	CPU-Timer Interrupt Enable. Reset type: SYSRSn 0h (R/W) = The CPU-Timer interrupt is disabled. 1h (R/W) = The CPU-Timer interrupt is enabled. If the timer decrements to zero, and TIE is set, the timer asserts its interrupt request.
13-12	RESERVED	R	0h	Reserved
11	FREE	R/W	0h	If the FREE bit is set to 1, then, upon a software breakpoint, the timer continues to run. If FREE is 0, then the SOFT bit controls the emulation behavior. Reset type: SYSRSn 0h (R/W) = Stop after the next decrement of the TIMH:TIM (hard stop) (SOFT bit controls the emulation behavior) 1h (R/W) = Free Run (SOFT bit is don't care, counter is free running)
10	SOFT	R/W	0h	If the FREE bit is set to 1, then, upon a software breakpoint, the timer continues to run (that is, free runs). In this case, SOFT is a don't care. But if FREE is 0, then SOFT takes effect. Reset type: SYSRSn 0h (R/W) = Stop after the next decrement of the TIMH:TIM (hard stop). (ONLY if FREE=0, if FREE=1 this bit is don't care) 1h (R/W) = Stop after the TIMH:TIM decrements to 0 (soft stop) In the SOFT STOP mode, the timer generates an interrupt before shutting down (since reaching 0 is the interrupt causing condition). (ONLY if FREE=0, if FREE=1 this bit is don't care)
9-6	RESERVED	R	0h	Reserved

Table 3-22. TCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	TRB	R/W	0h	Timer reload Reset type: SYSRSn 0h (R/W) = The TRB bit is always read as zero. Writes of 0 are ignored. 1h (R/W) = When you write a 1 to TRB, the TIMH:TIM is loaded with the value in the PRDH:PRD, and the prescaler counter (PSCH:PSC) is loaded with the value in the timer dividedown register (TDDRH:TDDR).
4	TSS	R/W	0h	CPU-Timer stop status bit. TSS is a 1-bit flag that stops or starts the CPU-timer. Reset type: SYSRSn 0h (R/W) = Reads of 0 indicate the CPU-timer is running. To start or restart the CPU-timer, set TSS to 0. At reset, TSS is cleared to 0 and the CPU-timer immediately starts. 1h (R/W) = Reads of 1 indicate that the CPU-timer is stopped. To stop the CPU-timer, set TSS to 1.
3-0	RESERVED	R	1h	Reserved

3.15.2.4 TPR Register (Offset = 6h) [Reset = 0000h]

TPR is shown in Figure 3-19 and described in Table 3-23.

Return to the Summary Table.

CPU-Timer, Prescale Register

Figure 3-19. TPR Register

15	14	13	12	11	10	9	8
PSC							
R-0h							
7	6	5	4	3	2	1	0
TDDR							
R/W-0h							

Table 3-23. TPR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	PSC	R	0h	CPU-Timer Prescale Counter. These bits hold the current prescale count for the timer. For every timer clock source cycle that the PSCH:PSC value is greater than 0, the PSCH:PSC decrements by one. One timer clock (output of the timer prescaler) cycle after the PSCH:PSC reaches 0, the PSCH:PSC is loaded with the contents of the TDDRH:TDDR, and the timer counter register (TIMH:TIM) decrements by one. The PSCH:PSC is also reloaded whenever the timer reload bit (TRB) is set by software. The PSCH:PSC can be checked by reading the register, but it cannot be set directly. It must get its value from the timer divide-down register (TDDRH:TDDR). At reset, the PSCH:PSC is set to 0. Reset type: SYSRSn
7-0	TDDR	R/W	0h	CPU-Timer Divide-Down. Every (TDDRH:TDDR + 1) timer clock source cycles, the timer counter register (TIMH:TIM) decrements by one. At reset, the TDDRH:TDDR bits are cleared to 0. To increase the overall timer count by an integer factor, write this factor minus one to the TDDRH:TDDR bits. When the prescaler counter (PSCH:PSC) value is 0, one timer clock source cycle later, the contents of the TDDRH:TDDR reload the PSCH:PSC, and the TIMH:TIM decrements by one. TDDRH:TDDR also reloads the PSCH:PSC whenever the timer reload bit (TRB) is set by software. Reset type: SYSRSn

3.15.2.5 TPRH Register (Offset = 7h) [Reset = 0000h]

TPRH is shown in Figure 3-20 and described in Table 3-24.

Return to the Summary Table.

CPU-Timer, Prescale Register High

Figure 3-20. TPRH Register

15	14	13	12	11	10	9	8
PSCH							
R-0h							
7	6	5	4	3	2	1	0
TDDRH							
R/W-0h							

Table 3-24. TPRH Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	PSCH	R	0h	See description of TIMERxTPR. Reset type: SYSRSn
7-0	TDDRH	R/W	0h	See description of TIMERxTPR. Reset type: SYSRSn

3.15.3 PIE_CTRL_REGS Registers

Table 3-25 lists the memory-mapped registers for the PIE_CTRL_REGS registers. All register offset addresses not listed in Table 3-25 should be considered as reserved locations and the register contents should not be modified.

Table 3-25. PIE_CTRL_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	PIECTRL	ePIE Control Register	
1h	PIEACK	Interrupt Acknowledge Register	
2h	PIEIER1	Interrupt Group 1 Enable Register	
3h	PIEIFR1	Interrupt Group 1 Flag Register	
4h	PIEIER2	Interrupt Group 2 Enable Register	
5h	PIEIFR2	Interrupt Group 2 Flag Register	
6h	PIEIER3	Interrupt Group 3 Enable Register	
7h	PIEIFR3	Interrupt Group 3 Flag Register	
8h	PIEIER4	Interrupt Group 4 Enable Register	
9h	PIEIFR4	Interrupt Group 4 Flag Register	
Ah	PIEIER5	Interrupt Group 5 Enable Register	
Bh	PIEIFR5	Interrupt Group 5 Flag Register	
Ch	PIEIER6	Interrupt Group 6 Enable Register	
Dh	PIEIFR6	Interrupt Group 6 Flag Register	
Eh	PIEIER7	Interrupt Group 7 Enable Register	
Fh	PIEIFR7	Interrupt Group 7 Flag Register	
10h	PIEIER8	Interrupt Group 8 Enable Register	
11h	PIEIFR8	Interrupt Group 8 Flag Register	
12h	PIEIER9	Interrupt Group 9 Enable Register	
13h	PIEIFR9	Interrupt Group 9 Flag Register	
14h	PIEIER10	Interrupt Group 10 Enable Register	
15h	PIEIFR10	Interrupt Group 10 Flag Register	
16h	PIEIER11	Interrupt Group 11 Enable Register	
17h	PIEIFR11	Interrupt Group 11 Flag Register	
18h	PIEIER12	Interrupt Group 12 Enable Register	
19h	PIEIFR12	Interrupt Group 12 Flag Register	

Complex bit access types are encoded to fit into small table cells. Table 3-26 shows the codes that are used for access types in this section.

Table 3-26. PIE_CTRL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		

Table 3-26. PIE_CTRL_REGS Access Type Codes (continued)

Access Type	Code	Description
-n		Value after reset or the default value

3.15.3.1 PIECTRL Register (Offset = 0h) [Reset = 0000h]

PIECTRL is shown in Figure 3-21 and described in Table 3-27.

Return to the Summary Table.

ePIE Control Register

Figure 3-21. PIECTRL Register

15	14	13	12	11	10	9	8
PIEVECT							
R-0h							
7	6	5	4	3	2	1	0
PIEVECT							ENPIE
R-0h							R/W-0h

Table 3-27. PIECTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	PIEVECT	R	0h	These bits indicate the vector address of the vector fetched from the ePIE vector table. The least significant bit of the address is ignored and only bits 1 to 15 of the address are shown. The vector value can be read by the user to determine which interrupt generated the vector fetch. Note: When a NMI is serviced, the PIEVECT bit-field does not reflect the vector as it does for other interrupts. Reset type: SYSRSn
0	ENPIE	R/W	0h	Enable vector fetching from ePIE block. This bit must be set to 1 for peripheral interrupts to work. All ePIE registers (PIEACK, PIEIFR, PIEIER) can be accessed even when the ePIE block is disabled. Reset type: SYSRSn

3.15.3.2 PIEACK Register (Offset = 1h) [Reset = 0000h]

PIEACK is shown in Figure 3-22 and described in Table 3-28.

Return to the Summary Table.

Acknowledge Register
When an interrupt propagates from the ePIE to a CPU interrupt line, the interrupt group's PIEACK bit is set. This prevents other interrupts in that group from propagating to the CPU while the first interrupt is handled. Writing a 1 to a PIEACK bit clears it and allows another interrupt from the corresponding group to propagate. ISRs for PIE interrupts should clear the group's PIEACK bit before returning from the interrupt.
Writes of 0 are ignored.

Figure 3-22. PIEACK Register

15	14	13	12	11	10	9	8
RESERVED				ACK12	ACK11	ACK10	ACK9
R-0-0h				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
ACK8	ACK7	ACK6	ACK5	ACK4	ACK3	ACK2	ACK1
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

Table 3-28. PIEACK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11	ACK12	R/W1S	0h	Acknowledge PIE Interrupt Group 12 Reset type: SYSRSn
10	ACK11	R/W1S	0h	Acknowledge PIE Interrupt Group 11 Reset type: SYSRSn
9	ACK10	R/W1S	0h	Acknowledge PIE Interrupt Group 10 Reset type: SYSRSn
8	ACK9	R/W1S	0h	Acknowledge PIE Interrupt Group 9 Reset type: SYSRSn
7	ACK8	R/W1S	0h	Acknowledge PIE Interrupt Group 8 Reset type: SYSRSn
6	ACK7	R/W1S	0h	Acknowledge PIE Interrupt Group 7 Reset type: SYSRSn
5	ACK6	R/W1S	0h	Acknowledge PIE Interrupt Group 6 Reset type: SYSRSn
4	ACK5	R/W1S	0h	Acknowledge PIE Interrupt Group 5 Reset type: SYSRSn
3	ACK4	R/W1S	0h	Acknowledge PIE Interrupt Group 4 Reset type: SYSRSn
2	ACK3	R/W1S	0h	Acknowledge PIE Interrupt Group 3 Reset type: SYSRSn
1	ACK2	R/W1S	0h	Acknowledge PIE Interrupt Group 2 Reset type: SYSRSn
0	ACK1	R/W1S	0h	Acknowledge PIE Interrupt Group 1 Reset type: SYSRSn

3.15.3.3 PIEIER1 Register (Offset = 2h) [Reset = 0000h]

PIEIER1 is shown in Figure 3-23 and described in Table 3-29.

Return to the Summary Table.

Interrupt Group 1 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-23. PIEIER1 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-29. PIEIER1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 1.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 1.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 1.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 1.1 Reset type: SYSRSn

3.15.3.4 PIEIFR1 Register (Offset = 3h) [Reset = 0000h]

PIEIFR1 is shown in Figure 3-24 and described in Table 3-30.

Return to the Summary Table.

Interrupt Group 1 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-24. PIEIFR1 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-30. PIEIFR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 1.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 1.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 1.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 1.1 Reset type: SYSRSn

3.15.3.5 PIEIER2 Register (Offset = 4h) [Reset = 0000h]

PIEIER2 is shown in Figure 3-25 and described in Table 3-31.

Return to the Summary Table.

Interrupt Group 2 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-25. PIEIER2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-31. PIEIER2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 2.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 2.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 2.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 2.1 Reset type: SYSRSn

3.15.3.6 PIEIFR2 Register (Offset = 5h) [Reset = 0000h]

PIEIFR2 is shown in Figure 3-26 and described in Table 3-32.

Return to the Summary Table.

Interrupt Group 2 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-26. PIEIFR2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-32. PIEIFR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 2.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 2.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 2.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 2.1 Reset type: SYSRSn

3.15.3.7 PIEIER3 Register (Offset = 6h) [Reset = 0000h]

PIEIER3 is shown in Figure 3-27 and described in Table 3-33.

Return to the Summary Table.

Interrupt Group 3 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-27. PIEIER3 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-33. PIEIER3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 3.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 3.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 3.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 3.1 Reset type: SYSRSn

3.15.3.8 PIEIFR3 Register (Offset = 7h) [Reset = 0000h]

PIEIFR3 is shown in Figure 3-28 and described in Table 3-34.

Return to the Summary Table.

Interrupt Group 3 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-28. PIEIFR3 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-34. PIEIFR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 3.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 3.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 3.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 3.1 Reset type: SYSRSn

3.15.3.9 PIEIER4 Register (Offset = 8h) [Reset = 0000h]

PIEIER4 is shown in Figure 3-29 and described in Table 3-35.

Return to the Summary Table.

Interrupt Group 4 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-29. PIEIER4 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-35. PIEIER4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 4.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 4.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 4.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 4.1 Reset type: SYSRSn

3.15.3.10 PIEIFR4 Register (Offset = 9h) [Reset = 0000h]

PIEIFR4 is shown in Figure 3-30 and described in Table 3-36.

Return to the Summary Table.

Interrupt Group 4 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-30. PIEIFR4 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-36. PIEIFR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 4.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 4.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 4.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 4.1 Reset type: SYSRSn

3.15.3.11 PIEIER5 Register (Offset = Ah) [Reset = 0000h]

PIEIER5 is shown in Figure 3-31 and described in Table 3-37.

Return to the Summary Table.

Interrupt Group 5 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-31. PIEIER5 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-37. PIEIER5 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 5.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 5.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 5.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 5.1 Reset type: SYSRSn

3.15.3.12 PIEIFR5 Register (Offset = Bh) [Reset = 0000h]

PIEIFR5 is shown in Figure 3-32 and described in Table 3-38.

Return to the Summary Table.

Interrupt Group 5 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-32. PIEIFR5 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-38. PIEIFR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 5.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 5.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 5.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 5.1 Reset type: SYSRSn

3.15.3.13 PIEIER6 Register (Offset = Ch) [Reset = 0000h]

PIEIER6 is shown in Figure 3-33 and described in Table 3-39.

Return to the Summary Table.

Interrupt Group 6 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-33. PIEIER6 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-39. PIEIER6 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 6.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 6.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 6.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 6.1 Reset type: SYSRSn

3.15.3.14 PIEIFR6 Register (Offset = Dh) [Reset = 0000h]

PIEIFR6 is shown in Figure 3-34 and described in Table 3-40.

Return to the Summary Table.

Interrupt Group 6 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-34. PIEIFR6 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-40. PIEIFR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 6.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 6.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 6.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 6.1 Reset type: SYSRSn

3.15.3.15 PIEIER7 Register (Offset = Eh) [Reset = 0000h]

PIEIER7 is shown in Figure 3-35 and described in Table 3-41.

Return to the Summary Table.

Interrupt Group 7 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-35. PIEIER7 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-41. PIEIER7 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 7.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 7.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 7.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 7.1 Reset type: SYSRSn

3.15.3.16 PIEIFR7 Register (Offset = Fh) [Reset = 0000h]

PIEIFR7 is shown in Figure 3-36 and described in Table 3-42.

Return to the Summary Table.

Interrupt Group 7 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-36. PIEIFR7 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-42. PIEIFR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 7.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 7.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 7.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 7.1 Reset type: SYSRSn

3.15.3.17 PIEIER8 Register (Offset = 10h) [Reset = 0000h]

PIEIER8 is shown in Figure 3-37 and described in Table 3-43.

Return to the Summary Table.

Interrupt Group 8 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-37. PIEIER8 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-43. PIEIER8 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 8.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 8.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 8.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 8.1 Reset type: SYSRSn

3.15.3.18 PIEIFR8 Register (Offset = 11h) [Reset = 0000h]

PIEIFR8 is shown in Figure 3-38 and described in Table 3-44.

Return to the Summary Table.

Interrupt Group 8 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-38. PIEIFR8 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-44. PIEIFR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 8.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 8.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 8.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 8.1 Reset type: SYSRSn

3.15.3.19 PIEIER9 Register (Offset = 12h) [Reset = 0000h]

PIEIER9 is shown in Figure 3-39 and described in Table 3-45.

Return to the Summary Table.

Interrupt Group 9 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-39. PIEIER9 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-45. PIEIER9 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 9.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 9.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 9.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 9.1 Reset type: SYSRSn

3.15.3.20 PIEIFR9 Register (Offset = 13h) [Reset = 0000h]

PIEIFR9 is shown in Figure 3-40 and described in Table 3-46.

Return to the Summary Table.

Interrupt Group 9 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-40. PIEIFR9 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-46. PIEIFR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 9.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 9.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 9.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 9.1 Reset type: SYSRSn

3.15.3.21 PIEIER10 Register (Offset = 14h) [Reset = 0000h]

PIEIER10 is shown in Figure 3-41 and described in Table 3-47.

Return to the Summary Table.

Interrupt Group 10 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-41. PIEIER10 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-47. PIEIER10 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 10.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 10.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 10.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 10.1 Reset type: SYSRSn

3.15.3.22 PIEIFR10 Register (Offset = 15h) [Reset = 0000h]

PIEIFR10 is shown in Figure 3-42 and described in Table 3-48.

Return to the Summary Table.

Interrupt Group 10 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-42. PIEIFR10 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-48. PIEIFR10 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 10.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 10.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 10.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 10.1 Reset type: SYSRSn

3.15.3.23 PIEIER11 Register (Offset = 16h) [Reset = 0000h]

PIEIER11 is shown in Figure 3-43 and described in Table 3-49.

Return to the Summary Table.

Interrupt Group 11 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-43. PIEIER11 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-49. PIEIER11 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 11.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 11.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 11.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 11.1 Reset type: SYSRSn

3.15.3.24 PIEIFR11 Register (Offset = 17h) [Reset = 0000h]

PIEIFR11 is shown in Figure 3-44 and described in Table 3-50.

Return to the Summary Table.

Interrupt Group 11 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-44. PIEIFR11 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-50. PIEIFR11 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 11.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 11.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 11.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 11.1 Reset type: SYSRSn

3.15.3.25 PIEIER12 Register (Offset = 18h) [Reset = 0000h]

PIEIER12 is shown in Figure 3-45 and described in Table 3-51.

Return to the Summary Table.

Interrupt Group 12 Enable Register
These register bits individually enable an interrupt within a group. They behave very much like the bits in the CPU interrupt enable register (IER).
Setting a bit to 1 allows the corresponding interrupt to propagate to the CPU.
Setting a bit to 0 prevents the corresponding interrupt from propagating. Note that a peripheral interrupt signal can still set the PIEIFR bit for the disabled interrupt.

Figure 3-45. PIEIER12 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-51. PIEIER12 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Enable for Interrupt 12.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Enable for Interrupt 12.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Enable for Interrupt 12.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Enable for Interrupt 12.1 Reset type: SYSRSn

3.15.3.26 PIEIFR12 Register (Offset = 19h) [Reset = 0000h]

PIEIFR12 is shown in Figure 3-46 and described in Table 3-52.

Return to the Summary Table.

Interrupt Group 12 Flag Register
These register bits indicate whether each interrupt in the group is currently pending. They behave very much like the bits in the CPU interrupt flag register (IFR).
When a peripheral sends an interrupt, the corresponding bit is set. This bit is cleared when the interrupt propagates to the CPU, at which point PIEACK is set.
NOTE: PIE IFR flags can be written to create software interrupts.
The IFR flag will be cleared on a write of zero. Hence, when the intent is to fire an interrupt it may cause inadvertent cancellation of other interrupts. It is recommended to use this only for testing or with extreme caution in the application code. Reading the PIE IFR registers is safe.

Figure 3-46. PIEIFR12 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				INTx4	INTx3	INTx2	INTx1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-52. PIEIFR12 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3	INTx4	R/W	0h	Flag for Interrupt 12.4 Reset type: SYSRSn
2	INTx3	R/W	0h	Flag for Interrupt 12.3 Reset type: SYSRSn
1	INTx2	R/W	0h	Flag for Interrupt 12.2 Reset type: SYSRSn
0	INTx1	R/W	0h	Flag for Interrupt 12.1 Reset type: SYSRSn

3.15.4 WD_REGS Registers

Table 3-53 lists the memory-mapped registers for the WD_REGS registers. All register offset addresses not listed in Table 3-53 should be considered as reserved locations and the register contents should not be modified.

Table 3-53. WD_REGS Registers

Offset	Acronym	Register Name	Write Protection
22h	SCSR	System Control & Status Register	EALLOW
23h	WDCNTR	Watchdog Counter Register	EALLOW
25h	WDKEY	Watchdog Reset Key Register	EALLOW
29h	WDCR	Watchdog Control Register	EALLOW
2Ah	WDWCR	Watchdog Windowed Control Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 3-54 shows the codes that are used for access types in this section.

Table 3-54. WD_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.4.1 SCSR Register (Offset = 22h) [Reset = 0005h]

SCSR is shown in Figure 3-47 and described in Table 3-55.

Return to the Summary Table.

System Control & Status Register

Figure 3-47. SCSR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					WDINTS	WDENINT	WDOVERRIDE
R-0-0h					R-1h	R/W-0h	R/W1S-1h

Table 3-55. SCSR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-3	RESERVED	R-0	0h	Reserved
2	WDINTS	R	1h	Watchdog Interrupt Status This bit indicates the state of the active-low watchdog interrupt signal (synchronized to SYSCLK). If the watchdog interrupt is used to wake the system from a low-power mode, then that mode should only be entered while this bit is high. Likewise, this bit must go high before the watchdog can be safely disabled and re-enabled. Reset type: SYSRSn
1	WDENINT	R/W	0h	Watchdog Interrupt Enable/Reset Disable This bit determines whether the watchdog triggers an interrupt (WAKE/WDOG) or a reset (WDRS) when the counter expires. Reset type: SYSRSn
0	WDOVERRIDE	R/W1S	1h	Watchdog Enable Lock Writing a 1 to this bit clears it and locks the WDDIS bit in the WDCR register. The bit will remain in this state until the next system reset. Reads of this bit return its current value. Writing a 0 to this bit has no effect. Reset type: SYSRSn

3.15.4.2 WDCNTR Register (Offset = 23h) [Reset = 0000h]

WDCNTR is shown in Figure 3-48 and described in Table 3-56.

Return to the Summary Table.

Watchdog Counter Register

Figure 3-48. WDCNTR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
WDCNTR							
R-0h							

Table 3-56. WDCNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-0	WDCNTR	R	0h	Watchdog Counter These bits contain the current value of the watchdog counter. This counter increments with each WDCLK (scaled SECCLK) cycle. If the counter overflows, either an interrupt or a reset is generated based on the value of the WDINTEN bit in the SCSR register. If the correct value is written to the WDKEY register, this counter is reset to zero. Reset type: IORSn

3.15.4.3 WDKEY Register (Offset = 25h) [Reset = 0000h]

WDKEY is shown in Figure 3-49 and described in Table 3-57.

Return to the Summary Table.

Watchdog Reset Key Register

Figure 3-49. WDKEY Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
WDKEY							
R/W-0h							

Table 3-57. WDKEY Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R-0	0h	Reserved
7-0	WDKEY	R/W	0h	Watchdog Counter Reset Writing 0x55 followed by 0xAA will cause the watchdog counter to reset to zero, preventing an overflow. Writing other values has no effect. Reads of this register return the value of the WDCR register. Reset type: IORSn

3.15.4.4 WDCR Register (Offset = 29h) [Reset = 0000h]

WDCR is shown in Figure 3-50 and described in Table 3-58.

Return to the Summary Table.

Watchdog Control Register

Figure 3-50. WDCR Register

15	14	13	12	11	10	9	8
RESERVED				WDPRECLKDIV			
R-0-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED	WDDIS	WDCHK			WDPS		
R/W1S-0h	R/W-0h	R-0/W-0h			R/W-0h		

Table 3-58. WDCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R-0	0h	Reserved
11-8	WDPRECLKDIV	R/W	0h	Watchdog Clock Pre-divider These bits determine the watchdog clock pre-divider, which is the first of the two dividers between SECCLK and the watchdog counter clock (WDCLK). The frequency of WDCLK is given by the formulas: PREDIVCLK = SECCLK / Pre-divider WDCLK = PREDIVCLK / Prescaler The watchdog reset or interrupt pulse is 512 SECCLK cycles long, so the counter period must be longer. To guarantee this, the product of the prescaler and pre-divider must be greater than or equal to four. The default pre-divider value is 512. 0h : PREDIVCLK = SECCLK / 512 1h : PREDIVCLK = SECCLK / 1024 2h : PREDIVCLK = SECCLK / 2048 3h : PREDIVCLK = SECCLK / 4096 4h : Reserved 5h : Reserved 6h : Reserved 7h : Reserved 8h : PREDIVCLK = SECCLK / 2 9h : PREDIVCLK = SECCLK / 4 Ah : PREDIVCLK = SECCLK / 8 Bh : PREDIVCLK = SECCLK / 16 Ch : PREDIVCLK = SECCLK / 32 Dh : PREDIVCLK = SECCLK / 64 Eh : PREDIVCLK = SECCLK / 128 Fh : PREDIVCLK = SECCLK / 256 Reset type: IORSn
7	RESERVED	R/W1S	0h	Reserved
6	WDDIS	R/W	0h	Watchdog Disable Setting this bit disables the watchdog module. Clearing this bit enables the watchdog module. This bit can be locked by the WDOVERRIDE bit in the SCSR register. The watchdog is enabled on reset. Reset type: IORSn
5-3	WDCHK	R-0/W	0h	Watchdog Check Bits During any write to this register, these bits must be written with the value 101 (binary). Writing any other value will immediately trigger the watchdog reset or interrupt. Reset type: IORSn

Table 3-58. WDCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	WDPS	R/W	0h	Watchdog Clock Prescaler These bits determine the watchdog clock prescaler, which is the second of the two dividers between SECCLK and the watchdog counter clock (WDCLK). The frequency of WDCLK is given by the formulas: PREDIVCLK = SECCLK / Pre-divider WDCLK = PREDIVCLK / Prescaler The watchdog reset or interrupt pulse is 512 SECCLK cycles long, so the counter period must be longer. To guarantee this, the product of the prescaler and pre-divider must be greater than or equal to four. The default prescaler value is 1. 0h : WDCLK = PREDIVCLK / 1 1h : WDCLK = PREDIVCLK / 1 2h : WDCLK = PREDIVCLK / 2 3h : WDCLK = PREDIVCLK / 4 4h : WDCLK = PREDIVCLK / 8 5h : WDCLK = PREDIVCLK / 16 6h : WDCLK = PREDIVCLK / 32 7h : WDCLK = PREDIVCLK / 64 Reset type: IORSn

3.15.4.5 WDWCR Register (Offset = 2Ah) [Reset = 0000h]

WDWCR is shown in Figure 3-51 and described in Table 3-59.

Return to the Summary Table.

Watchdog Windowed Control Register

Figure 3-51. WDWCR Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0-0h							R-0h
7	6	5	4	3	2	1	0
MIN							
R/W-0h							

Table 3-59. WDWCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8	RESERVED	R	0h	Reserved
7-0	MIN	R/W	0h	Watchdog Window Threshold These bits specify the lower limit of the watchdog counter reset window. If the counter is reset via the WDKEY register before the counter value reaches the value in this register, the watchdog immediately triggers a reset or interrupt. Reset type: IORSn

3.15.5 NMI_INTRUPT_REGS Registers

Table 3-60 lists the memory-mapped registers for the NMI_INTRUPT_REGS registers. All register offset addresses not listed in Table 3-60 should be considered as reserved locations and the register contents should not be modified.

Table 3-60. NMI_INTRUPT_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	NMICFG	NMI Configuration Register	EALLOW
1h	NMIFLG	NMI Flag Register (SYSRsn Clear)	
2h	NMIFLGCLR	NMI Flag Clear Register	EALLOW
3h	NMIFLGFRC	NMI Flag Force Register	EALLOW
4h	NMIWDCNT	NMI Watchdog Counter Register	
5h	NMIWDPRD	NMI Watchdog Period Register	EALLOW
6h	NMISHDFLG	NMI Shadow Flag Register	
7h	ERRORSTS	Error pin status	
8h	ERRORSTSCLR	ERRORSTS clear register	EALLOW
9h	ERRORSTSFRC	ERRORSTS force register	EALLOW
Ah	ERRORCTL	Error pin control register	EALLOW
Bh	ERRORLOCK	Lock register to Error pin registers.	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 3-61 shows the codes that are used for access types in this section.

Table 3-61. NMI_INTRUPT_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.5.1 NMICFG Register (Offset = 0h) [Reset = 0000h]

NMICFG is shown in Figure 3-52 and described in Table 3-62.

Return to the Summary Table.

NMI Configuration Register

Figure 3-52. NMICFG Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							NMIE
R-0-0h							R/W1S-0h

Table 3-62. NMICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R-0	0h	Reserved
0	NMIE	R/W1S	0h	When set to 1 any condition will generate an NMI interrupt to the C28 CPU and kick off the NMI watchdog counter. As part of boot sequence this bit should be set after the device security related initialization is complete. 0 NMI disabled 1 NMI enabled Reset type: SYSRSn

3.15.5.2 NMIFLG Register (Offset = 1h) [Reset = 0000h]

NMIFLG is shown in Figure 3-53 and described in Table 3-63.

Return to the Summary Table.

NMI Flag Register (SYSRsn Clear)

Figure 3-53. NMIFLG Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	SWERR	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	UNCERR	CLOCKFAIL	NMIINT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 3-63. NMIFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14	RESERVED	R	0h	Reserved
13	SWERR	R	0h	SW Error Force NMI Flag: This bit indicates if an NMI was forced through the NMIFLGFRC register. This bit can only be cleared by the user writing to the respective bit in the NMIFLGCLR register or by an SYSRSn reset: 0 No SW Error force Generated 1 SW Error NMI is forced by SW. No further NMI pulses are generated until this flag is cleared by the user. Reset type: SYSRSn
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	UNCERR	R	0h	Flash/RAM/ROM Uncorrectable Error NMI Flag: This bit indicates if an uncorrectable error occurred on a memory access (by any master) and that condition is latched. This bit can only be cleared by the user writing to the corresponding clear bit in the NMIFLGCLR register or by SYSRSn reset: 0,No uncorrectable error condition pending 1, uncorrectable error condition generated Reset type: SYSRSn
1	CLOCKFAIL	R	0h	Clock Fail Interrupt Flag: These bits indicates if the CLOCKFAIL condition is latched. These bits can only be cleared by the user writing to the respective bit in the NMIFLGCLR register or by SYSRSn reset: 0,No CLOCKFAIL Condition Pending 1,CLOCKFAIL Condition Generated Reset type: SYSRSn

Table 3-63. NMIFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	NMIINT	R	0h	NMI Interrupt Flag: This bit indicates if an NMI interrupt was generated. This bit can only be cleared by the user writing to the respective bit in the NMIFLGCLR register or by SYSRSn reset: 0 No NMI Interrupt Generated 1 NMI Interrupt Generated No further NMI interrupts pulses are generated until this flag is cleared by the user. Reset type: SYSRSn

3.15.5.3 NMIFLGCLR Register (Offset = 2h) [Reset = 0000h]

NMIFLGCLR is shown in Figure 3-54 and described in Table 3-64.

Return to the Summary Table.

NMI Flag Clear Register

Figure 3-54. NMIFLGCLR Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	SWERR	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	UNCERR	CLOCKFAIL	NMIINT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 3-64. NMIFLGCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13	SWERR	R-0/W1S	0h	Writing a 1 to the respective bit clears the corresponding flag bit in the NMIFLG and NMISHDFLG registers. Writes of 0 are ignored. Always reads back 0. Notes: [1] If hardware is trying to set a bit to 1 while software is trying to clear a bit to 0 on the same cycle, hardware has priority. [2] Users should clear the pending FAIL flag first and then clear the NMIINT flag. Reset type: SYSRSn
12	RESERVED	R-0/W1S	0h	Reserved
11	RESERVED	R-0/W1S	0h	Reserved
10	RESERVED	R-0/W1S	0h	Reserved
9	RESERVED	R-0/W1S	0h	Reserved
8	RESERVED	R-0/W1S	0h	Reserved
7	RESERVED	R-0/W1S	0h	Reserved
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	RESERVED	R-0/W1S	0h	Reserved
3	RESERVED	R-0/W1S	0h	Reserved
2	UNCERR	R-0/W1S	0h	Writing a 1 to the respective bit clears the corresponding flag bit in the NMIFLG and NMISHDFLG registers. Writes of 0 are ignored. Always reads back 0. Notes: [1] If hardware is trying to set a bit to 1 while software is trying to clear a bit to 0 on the same cycle, hardware has priority. [2] Users should clear the pending FAIL flag first and then clear the NMIINT flag. Reset type: SYSRSn
1	CLOCKFAIL	R-0/W1S	0h	Writing a 1 to the respective bit clears the corresponding flag bit in the NMIFLG and NMISHDFLG registers. Writes of 0 are ignored. Always reads back 0. Notes: [1] If hardware is trying to set a bit to 1 while software is trying to clear a bit to 0 on the same cycle, hardware has priority. [2] Users should clear the pending FAIL flag first and then clear the NMIINT flag. Reset type: SYSRSn

Table 3-64. NMIFLGCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	NMIINT	R-0/W1S	0h	Writing a 1 to the respective bit clears the corresponding flag bit in the NMIFLG and NMISHDFLG registers. Writes of 0 are ignored. Always reads back 0. Notes: [1] If hardware is trying to set a bit to 1 while software is trying to clear a bit to 0 on the same cycle, hardware has priority. [2] Users should clear the pending FAIL flag first and then clear the NMIINT flag. Reset type: SYSRSn

3.15.5.4 NMIFLGFRC Register (Offset = 3h) [Reset = 0000h]

NMIFLGFRC is shown in Figure 3-55 and described in Table 3-65.

Return to the Summary Table.

NMI Flag Force Register

Figure 3-55. NMIFLGFRC Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	SWERR	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	UNCERR	CLOCKFAIL	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0-0h

Table 3-65. NMIFLGFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13	SWERR	R-0/W1S	0h	Writing a 1 to these bits will set the respective FAIL flag in the NMIFLG and NMISHDFLG registers. Writes of 0 are ignored. Always reads back 0. This can be used as a means to test the NMI mechanisms. Reset type: SYSRSn
12	RESERVED	R-0/W1S	0h	Reserved
11	RESERVED	R-0/W1S	0h	Reserved
10	RESERVED	R-0/W1S	0h	Reserved
9	RESERVED	R-0/W1S	0h	Reserved
8	RESERVED	R-0/W1S	0h	Reserved
7	RESERVED	R-0/W1S	0h	Reserved
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	RESERVED	R-0/W1S	0h	Reserved
3	RESERVED	R-0/W1S	0h	Reserved
2	UNCERR	R-0/W1S	0h	Writing a 1 to these bits will set the respective FAIL flag in the NMIFLG and NMISHDFLG registers. Writes of 0 are ignored. Always reads back 0. This can be used as a means to test the NMI mechanisms. Reset type: SYSRSn
1	CLOCKFAIL	R-0/W1S	0h	Writing a 1 to these bits will set the respective FAIL flag in the NMIFLG and NMISHDFLG registers. Writes of 0 are ignored. Always reads back 0. This can be used as a means to test the NMI mechanisms. Reset type: SYSRSn
0	RESERVED	R-0	0h	Reserved

3.15.5.5 NMIWDCNT Register (Offset = 4h) [Reset = 0000h]

NMIWDCNT is shown in Figure 3-56 and described in Table 3-66.

Return to the Summary Table.

NMI Watchdog Counter Register

Figure 3-56. NMIWDCNT Register

15	14	13	12	11	10	9	8
NMIWDCNT							
R-0h							
7	6	5	4	3	2	1	0
NMIWDCNT							
R-0h							

Table 3-66. NMIWDCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	NMIWDCNT	R	0h	NMI Watchdog Counter: This 16-bit incremental counter will start incrementing whenever any one of the enabled FAIL flags are set. If the counter reaches the period value, an NMIRSn signal is fired which will then resets the system. The counter will reset to zero when it reaches the period value and will then restart counting if any of the enabled FAIL flags are set. If no enabled FAIL flag is set, then the counter will reset to zero and remain at zero until an enabled FAIL flag is set. Normally, the software would respond to the NMI interrupt generated and clear the offending FLAG(s) before the NMI watchdog triggers a reset. In some situations, the software may decide to allow the watchdog to reset the device anyway. The counter is clocked at the SYSCLKOUT rate. Reset type: SYSRSn

3.15.5.6 NMIWDPRD Register (Offset = 5h) [Reset = FFFFh]

NMIWDPRD is shown in Figure 3-57 and described in Table 3-67.

Return to the Summary Table.

NMI Watchdog Period Register

Figure 3-57. NMIWDPRD Register

15	14	13	12	11	10	9	8
NMIWDPRD							
R/W-FFFFh							
7	6	5	4	3	2	1	0
NMIWDPRD							
R/W-FFFFh							

Table 3-67. NMIWDPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	NMIWDPRD	R/W	FFFFh	NMI Watchdog Period: This 16-bit value contains the period value at which a reset is generated when the watchdog counter matches. At reset this value is set at the maximum. The software can decrease the period value at initialization time. Writing a PERIOD value that is smaller then the current counter value will automatically force an NMIRSn and hence reset the watchdog counter. Reset type: SYSRSn

3.15.5.7 NMISHDFLG Register (Offset = 6h) [Reset = 0000h]

NMISHDFLG is shown in Figure 3-58 and described in Table 3-68.

Return to the Summary Table.

NMI Shadow Flag Register

Figure 3-58. NMISHDFLG Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED	SWERR	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	UNCERR	CLOCKFAIL	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0-0h

Table 3-68. NMISHDFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R-0	0h	Reserved
14	RESERVED	R	0h	Reserved
13	SWERR	R	0h	Shadow NMI Flags: When an NMIFLG bit is set due to any of the possible NMI source in the device, the corresponding bit in this register is also set. Note that NMIFLGFRC and NMIFLGCLR register also affects the bits of this register in the same way as they do for the NMIFLG register. This register is resetted only by PORESETn. Notes: [1] This register is added to keep the definition of System Control Reset Cause Register Clean. Reset type: PORESETn
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	UNCERR	R	0h	Shadow NMI Flags: When an NMIFLG bit is set due to any of the possible NMI source in the device, the corresponding bit in this register is also set. Note that NMIFLGFRC and NMIFLGCLR register also affects the bits of this register in the same way as they do for the NMIFLG register. This register is resetted only by PORESETn. Notes: [1] This register is added to keep the definition of System Control Reset Cause Register Clean. Reset type: PORESETn
1	CLOCKFAIL	R	0h	Shadow NMI Flags: When an NMIFLG bit is set due to any of the possible NMI source in the device, the corresponding bit in this register is also set. Note that NMIFLGFRC and NMIFLGCLR register also affects the bits of this register in the same way as they do for the NMIFLG register. This register is resetted only by PORESETn. Notes: [1] This register is added to keep the definition of System Control Reset Cause Register Clean. Reset type: PORESETn

Table 3-68. NMISHDFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RESERVED	R-0	0h	Reserved

3.15.5.8 ERRORSTS Register (Offset = 7h) [Reset = 0000h]

ERRORSTS is shown in Figure 3-59 and described in Table 3-69.

Return to the Summary Table.

Error pin status

Figure 3-59. ERRORSTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						PINSTS	ERROR
R-0-0h						R-0h	R-0h

Table 3-69. ERRORSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R-0	0h	Reserved
1	PINSTS	R	0h	0, Error Pin is 0 1, Error Pin is 1 Reset type: PORESETn
0	ERROR	R	0h	0,None of the error sources were triggered. 1, One or more of the error sources triggered, or ERRORSTS.ERROR was set by a write of 1 to ERRORSTSFRC.ERROR bit. Once set, the ERROR flag can be cleared by writing 1 to ERRORSTSCLR.ERROR bit. Following are the events/triggers which can set this bit: 1. nmi interrupt on C28x 2. Watchdog reset 3. Error on a Pie vector fetch 4. Efuse error On a read of this bit, the pin Error pin state will be returned. Reset type: PORESETn

3.15.5.9 ERRORSTSCLR Register (Offset = 8h) [Reset = 0000h]

ERRORSTSCLR is shown in Figure 3-60 and described in Table 3-70.

Return to the Summary Table.

ERRORSTS clear register

Figure 3-60. ERRORSTSCLR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							ERROR
R-0-0h							R-0/W1S-0h

Table 3-70. ERRORSTSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R-0	0h	Reserved
0	ERROR	R-0/W1S	0h	0,No effect 1, ERRORSTS.ERROR is cleared to 0 Reset type: PORESETn

3.15.5.10 ERRORSTSFRC Register (Offset = 9h) [Reset = 0000h]

ERRORSTSFRC is shown in Figure 3-61 and described in Table 3-71.

Return to the Summary Table.

ERRORSTS force register

Figure 3-61. ERRORSTSFRC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							ERROR
R-0-0h							R-0/W1S-0h

Table 3-71. ERRORSTSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R-0	0h	Reserved
0	ERROR	R-0/W1S	0h	0,No effect 1, ERRORSTS.ERROR is set to 1 Reset type: PORESETn

3.15.5.11 ERRORCTL Register (Offset = Ah) [Reset = 0000h]

ERRORCTL is shown in Figure 3-62 and described in Table 3-72.

Return to the Summary Table.

Error pin control register

Figure 3-62. ERRORCTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							ERRORPOLSEL
R-0-0h							R/W-0h

Table 3-72. ERRORCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R-0	0h	Reserved
0	ERRORPOLSEL	R/W	0h	0, If ERRORSTS.ERROR is 1, Error pin will be driven with a value of 0, else 1. 1, If ERRORSTS.ERROR is 1, Error pin will be driven with a value of 1, else 0. Reset type: PORESETn

3.15.5.12 ERRORLOCK Register (Offset = Bh) [Reset = 0000h]

ERRORLOCK is shown in Figure 3-63 and described in Table 3-73.

Return to the Summary Table.

Lock register to Error pin registers.

Figure 3-63. ERRORLOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							ERRORCTL
R-0-0h							R/WSonce-0h

Table 3-73. ERRORLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R-0	0h	Reserved
0	ERRORCTL	R/WSonce	0h	0, Writes to ERRORCTL register allowed. 1, Writes to ERRORCTL register is blocked. Writes of 0 to this bit has no effect. Write of 1 will set this bit, cleared only on a SYSRSn. Reset type: SYSRSn

3.15.6 XINT_REGS Registers

Table 3-74 lists the memory-mapped registers for the XINT_REGS registers. All register offset addresses not listed in Table 3-74 should be considered as reserved locations and the register contents should not be modified.

Table 3-74. XINT_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	XINT1CR	XINT1 configuration register	
1h	XINT2CR	XINT2 configuration register	
2h	XINT3CR	XINT3 configuration register	
3h	XINT4CR	XINT4 configuration register	
4h	XINT5CR	XINT5 configuration register	
8h	XINT1CTR	XINT1 counter register	
9h	XINT2CTR	XINT2 counter register	
Ah	XINT3CTR	XINT3 counter register	

Complex bit access types are encoded to fit into small table cells. Table 3-75 shows the codes that are used for access types in this section.

Table 3-75. XINT_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.6.1 XINT1CR Register (Offset = 0h) [Reset = 0000h]

XINT1CR is shown in Figure 3-64 and described in Table 3-76.

Return to the Summary Table.

XINT1 configuration register

Figure 3-64. XINT1CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-76. XINT1CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.15.6.2 XINT2CR Register (Offset = 1h) [Reset = 0000h]

XINT2CR is shown in Figure 3-65 and described in Table 3-77.

Return to the Summary Table.

XINT2 configuration register

Figure 3-65. XINT2CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-77. XINT2CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.15.6.3 XINT3CR Register (Offset = 2h) [Reset = 0000h]

XINT3CR is shown in Figure 3-66 and described in Table 3-78.

Return to the Summary Table.

XINT3 configuration register

Figure 3-66. XINT3CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-78. XINT3CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.15.6.4 XINT4CR Register (Offset = 3h) [Reset = 0000h]

XINT4CR is shown in Figure 3-67 and described in Table 3-79.

Return to the Summary Table.

XINT4 configuration register

Figure 3-67. XINT4CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-79. XINT4CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.15.6.5 XINT5CR Register (Offset = 4h) [Reset = 0000h]

XINT5CR is shown in Figure 3-68 and described in Table 3-80.

Return to the Summary Table.

XINT5 configuration register

Figure 3-68. XINT5CR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

Table 3-80. XINT5CR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R-0	0h	Reserved
3-2	POLARITY	R/W	0h	00: Interrupt is selected as negative edge triggered 01: Interrupt is selected as positive edge triggered 10: Interrupt is selected as negative edge triggered 11: Interrupt is selected as positive or negative edge triggered Reset type: SYSRSn
1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	0: Interrupt Disabled 1: Interrupt Enabled Reset type: SYSRSn

3.15.6.6 XINT1CTR Register (Offset = 8h) [Reset = 0000h]

XINT1CTR is shown in Figure 3-69 and described in Table 3-81.

Return to the Summary Table.

XINT1 counter register

Figure 3-69. XINT1CTR Register

15	14	13	12	11	10	9	8
INTCTR							
R-0h							
7	6	5	4	3	2	1	0
INTCTR							
R-0h							

Table 3-81. XINT1CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	INTCTR	R	0h	This is a free running 16-bit up-counter that is clocked at the SYSCLKOUT rate. The counter value is reset to 0x0000 when a valid interrupt edge is detected and then continues counting until the next valid interrupt edge is detected. The counter must only be reset by the selected POLARITY edge as selected in the respective interrupt control register. When the interrupt is disabled, the counter will stop. The counter is a free-running counter and will wrap around to zero when the max value is reached. The counter is a read only register and can only be reset to zero by a valid interrupt edge or by reset. Reset type: SYSRSn

3.15.6.7 XINT2CTR Register (Offset = 9h) [Reset = 0000h]

XINT2CTR is shown in Figure 3-70 and described in Table 3-82.

Return to the Summary Table.

XINT2 counter register

Figure 3-70. XINT2CTR Register

15	14	13	12	11	10	9	8
INTCTR							
R-0h							
7	6	5	4	3	2	1	0
INTCTR							
R-0h							

Table 3-82. XINT2CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	INTCTR	R	0h	This is a free running 16-bit up-counter that is clocked at the SYSCLKOUT rate. The counter value is reset to 0x0000 when a valid interrupt edge is detected and then continues counting until the next valid interrupt edge is detected. The counter must only be reset by the selected POLARITY edge as selected in the respective interrupt control register. When the interrupt is disabled, the counter will stop. The counter is a free-running counter and will wrap around to zero when the max value is reached. The counter is a read only register and can only be reset to zero by a valid interrupt edge or by reset. Reset type: SYSRSn

3.15.6.8 XINT3CTR Register (Offset = Ah) [Reset = 0000h]

XINT3CTR is shown in Figure 3-71 and described in Table 3-83.

Return to the Summary Table.

XINT3 counter register

Figure 3-71. XINT3CTR Register

15	14	13	12	11	10	9	8
INTCTR							
R-0h							
7	6	5	4	3	2	1	0
INTCTR							
R-0h							

Table 3-83. XINT3CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	INTCTR	R	0h	This is a free running 16-bit up-counter that is clocked at the SYSCLKOUT rate. The counter value is reset to 0x0000 when a valid interrupt edge is detected and then continues counting until the next valid interrupt edge is detected. The counter must only be reset by the selected POLARITY edge as selected in the respective interrupt control register. When the interrupt is disabled, the counter will stop. The counter is a free-running counter and will wrap around to zero when the max value is reached. The counter is a read only register and can only be reset to zero by a valid interrupt edge or by reset. Reset type: SYSRSn

3.15.7 SYNC_SOC_REGS Registers

Table 3-84 lists the memory-mapped registers for the SYNC_SOC_REGS registers. All register offset addresses not listed in Table 3-84 should be considered as reserved locations and the register contents should not be modified.

Table 3-84. SYNC_SOC_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	SYNCSELECT	Sync Input and Output Select Register	EALLOW
2h	ADCSOCOUTSELECT	External ADCSOC Select Register	EALLOW
4h	SYNCSOCLOCK	SYNCSEL and EXTADCSOC Select Lock register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 3-85 shows the codes that are used for access types in this section.

Table 3-85. SYNC_SOC_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.7.1 SYNCSELECT Register (Offset = 0h) [Reset = 00000000h]

SYNCSELECT is shown in Figure 3-72 and described in Table 3-86.

Return to the Summary Table.

Sync Input and Output Select Register

Figure 3-72. SYNCSELECT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED			SYNCOUT					RESERVED							
R-0-0h			R/W-0h					R-0-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															
R-0-0h															

Table 3-86. SYNCSELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R-0	0h	Reserved
28-24	SYNCOUT	R/W	0h	Select Syncout Source: 00000: PWM1SYNCOUT selected to drive the EXTSYNCOUT pin. 00001: PWM2SYNCOUT selected to drive the EXTSYNCOUT pin (reserved). 00010: PWM3SYNCOUT selected to drive the EXTSYNCOUT pin. 00011: PWM4SYNCOUT selected to drive the EXTSYNCOUT pin (reserved). 00100: PWM5SYNCOUT selected to drive the EXTSYNCOUT pin (reserved). 00101: PWM6SYNCOUT selected to drive the EXTSYNCOUT pin (reserved). 00110: Reserved 00111: Reserved 01000: Reserved 01001: Reserved 01010: Reserved 01011: Reserved 01100: Reserved 01101: Reserved 01110: Reserved 01111: Reserved 10000: Reserved 10001: Reserved 10010: Reserved 10011: Reserved 10100: Reserved 10101: Reserved 10110: Reserved 10111: Reserved 11000: ECAP1SYNCOUT selected to drive the EXTSYNCOUT pin. 11001: ECAP2SYNCOUT selected to drive the EXTSYNCOUT pin (reserved). 11010: Reserved 11011: Reserved 11100: Reserved 11101: Reserved 11110: Reserved 11111: Reserved Notes: [1] Reserved position defaults to 00 selection Reset type: SYSRSn
23-0	RESERVED	R-0	0h	Reserved

3.15.7.2 ADCSOCOUTSELECT Register (Offset = 2h) [Reset = 00000000h]

ADCSOCOUTSELECT is shown in Figure 3-73 and described in Table 3-87.

Return to the Summary Table.

External ADCSOC Select Register

Figure 3-73. ADCSOCOUTSELECT Register

31	30	29	28	27	26	25	24
RESERVED				RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	PWM3SOCBEN	RESERVED	PWM1SOCBEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED				RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	PWM3SOCAEN	RESERVED	PWM1SOCAEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-87. ADCSOCOUTSELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	PWM3SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective PWM SOCB output is not selected 1: Respective PWM SOCB output is selected Reset type: SYSRSn
17	RESERVED	R/W	0h	Reserved
16	PWM1SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective PWM SOCB output is not selected 1: Respective PWM SOCB output is selected Reset type: SYSRSn
15-12	RESERVED	R-0	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved

Table 3-87. ADCSOCOUTSELECT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	PWM3SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective PWM SOCA output is not selected 1: Respective PWM SOCA output is selected Reset type: SYSRSn
1	RESERVED	R/W	0h	Reserved
0	PWM1SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective PWM SOCA output is not selected 1: Respective PWM SOCA output is selected Reset type: SYSRSn

3.15.7.3 SYNCSOCLOCK Register (Offset = 4h) [Reset = 00000000h]

SYNCSOCLOCK is shown in Figure 3-74 and described in Table 3-88.

Return to the Summary Table.

SYNCSEL and EXTADCSOC Select Lock register

Figure 3-74. SYNCSOCLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						ADCSOCOUTSELECT	SYNCSELECT
R-0-0h						R/WSonce-0h	R/WSonce-0h

Table 3-88. SYNCSOCLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	ADCSOCOUTSELECT	R/WSonce	0h	ADCSOCOUTSELECT Register Lock bit: 0: Respective register is not locked 1: Respective register is locked. Notes: [1] Any bit in this register, once set can only be creaed through a SYSRSn. Write of 0 to any bit of this regtister has no effect [2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed Reset type: SYSRSn
0	SYNCSELECT	R/WSonce	0h	SYNCSELECT Register Lock bit: 0: Respective register is not locked 1: Respective register is locked. Notes: [1] Any bit in this register, once set can only be creaed through a SYSRSn. Write of 0 to any bit of this regtister has no effect [2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed Reset type: SYSRSn

3.15.8 DMA_CLA_SRC_SEL_REGS Registers

Table 3-89 lists the memory-mapped registers for the DMA_CLA_SRC_SEL_REGS registers. All register offset addresses not listed in Table 3-89 should be considered as reserved locations and the register contents should not be modified.

Table 3-89. DMA_CLA_SRC_SEL_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	DMACLASRCSELLOCK	DMA/CLA Trigger Source Select Lock Register	EALLOW
2h	DMACHSRCSEL1	DMA Channel Trigger Source Select Register-1	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 3-90 shows the codes that are used for access types in this section.

Table 3-90. DMA_CLA_SRC_SEL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.8.1 DMACLASRCSELLOCK Register (Offset = 0h) [Reset = 00000000h]

DMACLASRCSELLOCK is shown in Figure 3-75 and described in Table 3-91.

Return to the Summary Table.

DMA/CLA Trigger Source Select Lock Register

Figure 3-75. DMACLASRCSELLOCK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ALL
R-0-0h															R/WSonce-0h

Table 3-91. DMACLASRCSELLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	ALL	R/WSonce	0h	All DMACLASRCSEL Registers Lock bit: 0: Respective register is not locked 1: Respective register is locked. Notes: [1] Any SOnce bit in this register, once set can only be cleared through a SYSRSn. Write of 0 to any bit of this register has no effect [2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed Reset type: SYSRSn

3.15.8.2 DMACHSRCSEL1 Register (Offset = 2h) [Reset = 00000000h]

DMACHSRCSEL1 is shown in Figure 3-76 and described in Table 3-92.

Return to the Summary Table.

DMA Channel Trigger Source Select Register-1

Figure 3-76. DMACHSRCSEL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED		CH2					
R-0-0h		R/W-0h					
7	6	5	4	3	2	1	0
RESERVED		CH1					
R-0-0h		R/W-0h					

Table 3-92. DMACHSRCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	Reserved
23-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R-0	0h	Reserved
13-8	CH2	R/W	0h	Selects the Trigger and Sync Source CH2 of DMA Reset type: SYSRSn
7-6	RESERVED	R-0	0h	Reserved
5-0	CH1	R/W	0h	Selects the Trigger and Sync Source CH1 of DMA Reset type: SYSRSn

3.15.9 DEV_CFG_REGS Registers

Table 3-93 lists the memory-mapped registers for the DEV_CFG_REGS registers. All register offset addresses not listed in Table 3-93 should be considered as reserved locations and the register contents should not be modified.

Table 3-93. DEV_CFG_REGS Registers

Offset	Acronym	Register Name	Write Protection
2h	PARTIDL	Lower 32-bit of Device PART Identification Number	
4h	PARTIDH	Upper 32-bit of Device PART Identification Number	
6h	REVID	Device Revision Number	
Eh	DC_MEMORY	Device Capability: Memory Blocks Customization	
10h	PERCNF	Peripheral Configuration register - GPIO	
12h	TRIMERRSTS	TRIM Error Status register	
1Eh	SOFTPRES_PROC_INFRA	Processing and Infra Blocks Software Reset register	EALLOW
20h	SOFTPRES_CTRL_PERIPH	Control Peripherals Software Reset register	EALLOW
22h	SOFTPRES_COMM_PERIPH	Communication Peripherals Software Reset register	EALLOW
24h	SOFTPRES_JTAG	JTAG Software Reset register	EALLOW
28h	TAP_STATUS	Status of JTAG State machine & Debugger Connect	
2Ah	ECAPTYPE	Configures ECAP Type for the device	EALLOW
2Ch	TAP_CONTROL	Disable TAP control	

Complex bit access types are encoded to fit into small table cells. Table 3-94 shows the codes that are used for access types in this section.

Table 3-94. DEV_CFG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
WOnce	W Once	Write Write once
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.9.1 PARTIDL Register (Offset = 2h) [Reset = 0000XXX0h]

PARTIDL is shown in Figure 3-77 and described in Table 3-95.

Return to the Summary Table.

Lower 32-bit of Device PART Identification Number

Figure 3-77. PARTIDL Register

31	30	29	28	27	26	25	24
PARTID_FORMAT_REV				RESERVED			
R-0h				R-0-0h			
23	22	21	20	19	18	17	16
FLASH_SIZE							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED			
R-0h	R-Xh	R-0h	R-Xh	R-Xh			
7	6	5	4	3	2	1	0
QUAL		RESERVED	RESERVED			RESERVED	
R-Xh		R-0h	R-0h			R-0h	

Table 3-95. PARTIDL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	PARTID_FORMAT_REV	R	0h	PARTID_FORMAT_REV Reset type: PORESETn
27-24	RESERVED	R-0	0h	Reserved
23-16	FLASH_SIZE	R/W	0h	0x0 - Reserved 0x1 - 32 KB 0x2 - 64 KB 0x3 - 96 KB 0x4 - 128 KB Others - Reserved Reset type: PORESETn
15	RESERVED	R	0h	Reserved
14	RESERVED	R	Xh	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	Xh	Reserved
11-8	RESERVED	R	Xh	Reserved
7-6	QUAL	R	Xh	0 = Engineering sample (TMX) 1 = Pilot production (TMP) 2 = Fully qualified (TMS) Reset type: PORESETn
5	RESERVED	R	0h	Reserved
4-3	RESERVED	R	0h	Reserved
2-0	RESERVED	R	0h	Reserved

3.15.9.2 PARTIDH Register (Offset = 4h) [Reset = 10XX0500h]

PARTIDH is shown in Figure 3-78 and described in Table 3-96.

Return to the Summary Table.

Upper 32-bit of Device PART Identification Number

Figure 3-78. PARTIDH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEVICE_CLASS_ID								PARTNO							
R-10h								R-XXh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAMILY								RESERVED				RESERVED			
R-5h								R-0h				R-0h			

Table 3-96. PARTIDH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DEVICE_CLASS_ID	R	10h	Device class ID Device Value Sonata 0x40 Aumento 0x50 SOPRANO 0x00 POTENZA 0x01 SOPRANO-FELICE 0x02 TENOR 0x03 TOPOLINO 0x04 TOPOGRANDE 0X05 TOPOARIA 0x06 TOPOAUTO 0x07 GATTINO 0x08 Predator 0x09 Gara 0x0C Veloce 0x0D Invicta-MC1 0x10 Reset type: PORESETn
23-16	PARTNO	R	XXh	Refer to Datasheet for Device Part Number Reset type: PORESETn
15-8	FAMILY	R	5h	Device Family This field categorizes the device to one of the C2000 device families, namely Delfino, Piccolo (Harmony class), Delfino-Single core, Concerto etc. Reset type: PORESETn
7-4	RESERVED	R	0h	Reserved
3-0	RESERVED	R	0h	Reserved

3.15.9.3 REVID Register (Offset = 6h) [Reset = 00000000h]

REVID is shown in Figure 3-79 and described in Table 3-97.

Return to the Summary Table.

Device Revision Number

Figure 3-79. REVID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REVID															
R-0-0h																R/WOnce-0h															

Table 3-97. REVID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	REVID	R/WOnce	0h	Device Revision ID. Loaded from flash trim sector by boot rom. Reset value is die-specific. Reset type: XRSn

3.15.9.4 DC_MEMORY Register (Offset = Eh) [Reset = 00000X0Xh]

DC_MEMORY is shown in Figure 3-80 and described in Table 3-98.

Return to the Summary Table.

Device Capability: Memory Blocks Customization

Figure 3-80. DC_MEMORY Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	BANK1_32KB_4	BANK1_32KB_3	BANK1_32KB_2	BANK1_32KB_1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-Xh
7	6	5	4	3	2	1	0
BANK0_32KB_8	BANK0_32KB_7	BANK0_32KB_6	BANK0_32KB_5	BANK0_32KB_4	BANK0_32KB_3	BANK0_32KB_2	BANK0_32KB_1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-Xh	R-Xh	R-Xh	R-Xh

Table 3-98. DC_MEMORY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	BANK1_32KB_4	R/W	0h	Flash Bank-1: Fourth 32 KB (upto 128 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
10	BANK1_32KB_3	R/W	0h	Flash Bank-1: Third 32 KB (upto 96 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
9	BANK1_32KB_2	R/W	0h	Flash Bank-1: Second 32 KB (upto 64 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
8	BANK1_32KB_1	R	Xh	Flash Bank-1: First 32 KB 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
7	BANK0_32KB_8	R/W	0h	Flash Bank-0: Eigth 32 KB (upto 256 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

Table 3-98. DC_MEMORY Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	BANK0_32KB_7	R/W	0h	Flash Bank-0: Seventh 32 KB (upto 224 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
5	BANK0_32KB_6	R/W	0h	Flash Bank-0: Sixth 32 KB (upto 192 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
4	BANK0_32KB_5	R/W	0h	Flash Bank-0: Fifth 32 KB (upto 160 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
3	BANK0_32KB_4	R	Xh	Flash Bank-0: Fourth 32 KB (upto 128 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
2	BANK0_32KB_3	R	Xh	Flash Bank-0: Third 32 KB (upto 96 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
1	BANK0_32KB_2	R	Xh	Flash Bank-0: Second 32 KB (upto 64 KB) 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn
0	BANK0_32KB_1	R	Xh	Flash Bank-0: First 32 KB 0: Respective sectors are not present in the device 1: Respective sectors are present in the device Reset type: PORESETn

3.15.9.5 PERCNF Register (Offset = 10h) [Reset = 0000000Xh]

PERCNF is shown in Figure 3-81 and described in Table 3-99.

Return to the Summary Table.

Peripheral Configuration register - GPIO

Figure 3-81. PERCNF Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						GPIO_230_227	GPIO_228_226
R-0-0h						R-Xh	R-Xh

Table 3-99. PERCNF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	GPIO_230_227	R	Xh	This bit is used to provide protection (i.e. avoid contention when both the pads are configured in output mode) for the packages when GPIO228 and GPIO226 are double bonded. 0: No protection. 1: GPIO227GZ will be forced to input mode when GPIO230 is configured as output Reset type: PORESETn
0	GPIO_228_226	R	Xh	This bit is used to provide protection (i.e. avoid contention when both the pads are configured in output mode) for the packages when GPIO228 and GPIO226 are double bonded. 0: No protection. 1: GPIO226GZ will be forced to input mode when GPIO228 is configured as output Reset type: PORESETn

3.15.9.6 TRIMERRSTS Register (Offset = 12h) [Reset = 00000000h]

TRIMERRSTS is shown in Figure 3-82 and described in Table 3-100.

Return to the Summary Table.

TRIM Error Status register

Figure 3-82. TRIMERRSTS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LERR															
R-0-0h																R/WSonce-0h															

Table 3-100. TRIMERRSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	LERR	R/WSonce	0h	TRIM information load error status. This will include error during SRAM repair also. 0x1: Correctable single bit error 0x2: Uncorrectable double bit error 0x20: Trim over timeout error Other: Non zero value indicates error during load Note: [1] This bit is updated by software. Details will be filled in once the Boot ROM related requirements are complete. It should have bits to indicate (i) Double bit error during trim load (ii) Single bit error during trim load (iii) Double bit error during SRAM repair load (iv) Single bit error error during SRAM repair load (v) SRAM repair error load (chain is broken) (vi) PWRUPSTS.TRIMOVER signal is not asserted even after the full wait time Reset type: XRSn

3.15.9.7 SOFTPRES_PROC_INFRA Register (Offset = 1Eh) [Reset = 0000000Xh]

SOFTPRES_PROC_INFRA is shown in Figure 3-83 and described in Table 3-101.

Return to the Summary Table.

When bits in this register are set, the respective module is in reset. All design data is lost and the module registers are returned to their reset states. Bits must be manually cleared after being set.

Figure 3-83. SOFTPRES_PROC_INFRA Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						FLASHA	DCC1
R-0h						R-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-Xh

Table 3-101. SOFTPRES_PROC_INFRA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9	FLASHA	R	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
8	DCC1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R	Xh	Reserved

3.15.9.8 SOFTPRES_CTRL_PERIPH Register (Offset = 20h) [Reset = 00XXXX0Xh]

SOFTPRES_CTRL_PERIPH is shown in Figure 3-84 and described in Table 3-102.

Return to the Summary Table.

When bits in this register are set, the respective module is in reset. All design data is lost and the module registers are returned to their reset states. Bits must be manually cleared after being set.

Figure 3-84. SOFTPRES_CTRL_PERIPH Register

31	30	29	28	27	26	25	24
RESERVED						RESERVED	PGA1
R-0-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	CMPSS3	CMPSS2	CMPSS1	RESERVED
R/W-0h	R/W-0h	R-Xh	R-Xh	R/W-0h	R-Xh	R-Xh	R-Xh
15	14	13	12	11	10	9	8
ADC_A	EQEP1	RESERVED	ECAP1	RESERVED	RESERVED	RESERVED	RESERVED
R-Xh	R-Xh	R-Xh	R-Xh	R/W-0h	R/W-0h	R-Xh	R-Xh
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	PWM3	RESERVED	PWM1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-Xh

Table 3-102. SOFTPRES_CTRL_PERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R-0	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	PGA1	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R	Xh	Reserved
20	RESERVED	R	Xh	Reserved
19	CMPSS3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
18	CMPSS2	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
17	CMPSS1	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
16	RESERVED	R	Xh	Reserved
15	ADC_A	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
14	EQEP1	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
13	RESERVED	R	Xh	Reserved

Table 3-102. SOFTPRES_CTRL_PERIPH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	ECAP1	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R	Xh	Reserved
8	RESERVED	R	Xh	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	PWM3	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
1	RESERVED	R/W	0h	Reserved
0	PWM1	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn

3.15.9.9 SOFTPRES_COMM_PERIPH Register (Offset = 22h) [Reset = 0000000Xh]

SOFTPRES_COMM_PERIPH is shown in Figure 3-85 and described in Table 3-103.

Return to the Summary Table.

When bits in this register are set, the respective module is in reset. All design data is lost and the module registers are returned to their reset states. Bits must be manually cleared after being set.

Figure 3-85. SOFTPRES_COMM_PERIPH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	SPI_A	RESERVED	UART_A	SCI_B	SCI_A	RESERVED	I2C_A
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-Xh

Table 3-103. SOFTPRES_COMM_PERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	SPI_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
5	RESERVED	R/W	0h	Reserved
4	UART_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
3	SCI_B	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
2	SCI_A	R/W	0h	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn
1	RESERVED	R/W	0h	Reserved
0	I2C_A	R	Xh	1: Module is under reset 0: Module reset is determined by the normal device reset structure Reset type: SYSRSn

3.15.9.10 SOFTPRES_JTAG Register (Offset = 24h) [Reset = 00000000h]

SOFTPRES_JTAG is shown in Figure 3-86 and described in Table 3-104.

Return to the Summary Table.

The Reset bit in this register needs to be set along with valid Key to ensure that JTAG nTRST is asserted. This is auto clear register.

Figure 3-86. SOFTPRES_JTAG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JTAG_nTRST_Key															
R-0/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												JTAG_nTRST			
R-0-0h												R/W-0h			

Table 3-104. SOFTPRES_JTAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	JTAG_nTRST_Key	R-0/W	0h	0xdcaf : Writing this Key value along with 0xA in JTAG_nTRST field causes a JTAG nTRST pulse generated to the JTAG state machine. Any other write does not have impact on the JTAG state machine, bits are self clear when Reset is asserted to JTAG state machine. Reset type: SYSRSn, TRSTn
15-4	RESERVED	R-0	0h	Reserved
3-0	JTAG_nTRST	R/W	0h	1010: Writing '1010' along with valid key in JTAG_nTRST_Key takes JTAG TAP to TLR state. Writing any other value or mismatched key does not have any effect on the JTAG TAP reset behavior. Once Reset to JTAG domain is asserted then this field is reset back to 0. Reset type: SYSRSn, TRSTn

3.15.9.11 TAP_STATUS Register (Offset = 28h) [Reset = 00000000h]

TAP_STATUS is shown in Figure 3-87 and described in Table 3-105.

Return to the Summary Table.

Status of JTAG State machine & Debugger Connect

Figure 3-87. TAP_STATUS Register

31	30	29	28	27	26	25	24
DCON	RESERVED						
R-0h	R-0-0h						
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
TAP_STATE							
R-0h							
7	6	5	4	3	2	1	0
TAP_STATE							
R-0h							

Table 3-105. TAP_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DCON	R	0h	DebugConnect indication from IcePick. Reset type: PORESETn
30-16	RESERVED	R-0	0h	Reserved
15-0	TAP_STATE	R	0h	TAP State Vector. With bits representing, Connect coresponding POTAP* output to the 0:TLR, 1:IDLE, 2:SELECTDR, 3:CAPDR, 4:SHIFTDR, 5:EXIT1DR, 6:PAUSEDR, 7:EXIT2DR, 8:UPDTDR, 9:SLECTIR, 10:CAPIR, 11:SHIFTIR, 12:EXIT1IR, 13:PAUSEIR, 14:EXIT2IR, 15:UPDTIR, Reset type: PORESETn

3.15.9.12 ECAPTYPE Register (Offset = 2Ah) [Reset = 00000000h]

ECAPTYPE is shown in Figure 3-88 and described in Table 3-106.

Return to the Summary Table.

Based on the configuration enables disables features associated with the ECAP type.

Figure 3-88. ECAPTYPE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
LOCK	RESERVED						
R/WSonce-0h	R-0-0h						
7	6	5	4	3	2	1	0
RESERVED						TYPE	
R-0-0h						R/W-0h	

Table 3-106. ECAPTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	LOCK	R/WSonce	0h	1: Write to this register is not allowed. 0: Write to this register is allowed. Reset type: SYSRSn
14-2	RESERVED	R-0	0h	Reserved
1-0	TYPE	R/W	0h	'00,10,11' : 1. No EALLOW protection to ECAP registers. '01' : 1. ECAP registers are EALLOW protected. Reset type: SYSRSn

3.15.9.13 TAP_CONTROL Register (Offset = 2Ch) [Reset = 00000000h]

TAP_CONTROL is shown in Figure 3-89 and described in Table 3-107.

Return to the Summary Table.

Disable TAP control

Figure 3-89. TAP_CONTROL Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							BSCAN_DIS
R-0-0h							R/W-0h

Table 3-107. TAP_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write to this register succeeds only if this field is written with a value of 0xa5a5 Note: [1] Due to this KEY, only 32-bit writes will succeed (provided the KEY matches). 16-bit writes to the upper or lower half of this register will be ignored Reset type: PORESETn
15-1	RESERVED	R-0	0h	Reserved
0	BSCAN_DIS	R/W	0h	Disables BSCAN TAP control : 0: BSCAN TAP control enabled 1: BSCAN TAP control disabled Reset type: PORESETn

3.15.10 CLK_CFG_REGS Registers

Table 3-108 lists the memory-mapped registers for the CLK_CFG_REGS registers. All register offset addresses not listed in Table 3-108 should be considered as reserved locations and the register contents should not be modified.

Table 3-108. CLK_CFG_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	CLKCFGLOCK	Lock bit for CLKCFG registers	EALLOW
2h	CLKSRCCTL1	Clock Source Control register-1	EALLOW
6h	CLKSRCCTL3	Clock Source Control register-3	EALLOW
8h	SYSPLLCTL	SYSPLL Control register	EALLOW
Ah	SYSPLLMULT	SYSPLL Multiplier register	EALLOW
Ch	SYSPLLSTS	SYSPLL Status register	
Eh	SYSCLKDIVSEL	System Clock Divider Select register	EALLOW
12h	XCLKOUTDIVSEL	XCLKOUT Divider Select register	EALLOW
14h	LOSPCP	Low Speed Clock Source Prescalar	EALLOW
16h	MCDCR	Missing Clock Detect Control Register	EALLOW
18h	X1CNT	10-bit Counter on X1 Clock	
1Ah	XTALCR	XTAL Control Register	EALLOW
1Ch	XTALCR2	XTAL Control Register for pad init	EALLOW
1Eh	CLKFAILCFG	Clock Fail cause Configuration	EALLOW
20h	CLKSRCSTS	Clock Source Status	

Complex bit access types are encoded to fit into small table cells. Table 3-109 shows the codes that are used for access types in this section.

Table 3-109. CLK_CFG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.10.1 CLKCFGLOCK Register (Offset = 0h) [Reset = 00000000h]

CLKCFGLOCK is shown in Figure 3-90 and described in Table 3-110.

Return to the Summary Table.

Lock bit for CLKCFG registers
Notes:
[1] Any bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect
[2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed

Figure 3-90. CLKCFGLOCK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ALL
R-0-0h															R/WSonce-0h

Table 3-110. CLKCFGLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	ALL	R/WSonce	0h	Lock bit for all CLKCFG registers 0: Respective register is not locked 1: Respective register is locked. Reset type: SYSRSn

3.15.10.2 CLKSRCCTL1 Register (Offset = 2h) [Reset = 00000002h]

CLKSRCCTL1 is shown in Figure 3-91 and described in Table 3-111.

Return to the Summary Table.

Clock Source Control register-1
This memory mapped register requires a delay of 45 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 45 NOP instructions.

Figure 3-91. CLKSRCCTL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		WDHALTI	RESERVED		SECCLKSRCSEL	OSCCLKSRCSEL	
R-0-0h		R/W-0h	R/W-0h		R/W-0h	R/W-2h	

Table 3-111. CLKSRCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	WDHALTI	R/W	0h	Watchdog HALT Mode Ignore Bit: This bit determines if WD is functional in the HALT mode or not. Writing to this bit will unlock the PLL and clear the SYSPLLSTS.LOCKS bit. 0 = WD is not functional in the HALT mode. Clock to WD is gated when system enters HALT mode. 1 = WD is functional in the HALT mode. Clock to WD is not gated Reset type: XRSn
4-3	RESERVED	R/W	0h	Reserved
2	SECCLKSRCSEL	R/W	0h	Secondary Clock Source Select Bit: This bit selects the source for SECCLK. 0 = WROSCBY8 (default on reset) 1 = SYSOSCBY4 Notes: When XTAL is selected as the OSCCLK source, the SECCLK can be chosen to use SYSOSCBY4, which is a more accurate clock compared to WROSCBY8 Reset type: XRSn

Table 3-111. CLKSRCCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	OSCCLKSRCSEL	R/W	2h	Oscillator Clock Source Select Bits: These bits select the source for OSCCLK. 00 = SYSOSCBY4 01 = External Oscillator (XTAL) 10 = WROSCBY8 (default) 11 = reserved (default to WROSCBY8) The user must wait 10 OSCCLK cycles before writing to SYSPLLMULT or disabling the previous clock source to allow the change to complete.. Notes: [1] WROSCBY8 is recommended to be used only after missing clock detection. Though not recommended due to frequency instability, if user wants to re-lock the PLL with WROSCBY8 (the back-up clock source) after missing clock is detected, he can do a MCLKCLR and lock the PLL. Reset type: XRSn

3.15.10.3 CLKSRCCTL3 Register (Offset = 6h) [Reset = 00000000h]

CLKSRCCTL3 is shown in Figure 3-92 and described in Table 3-112.

Return to the Summary Table.

Clock Source Control register-3
This memory mapped register requires a delay of 45 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 45 NOP instructions.

Figure 3-92. CLKSRCCTL3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												XCLKOUTSEL			
R-0-0h												R/W-0h			

Table 3-112. CLKSRCCTL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3-0	XCLKOUTSEL	R/W	0h	XCLKOUT Source Select Bit: These bits select the source for XCLKOUT: 0x0 = PLLSYSCLK (default on reset) 0x1 = PLLCLK 0x2 = SYSCLK 0x3 = WROSC 0x4 = SYSOSC 0x5 = WROSCBY8 0x6 = SYSOSCBY4 0x7 = XTAL OSC o/p clock 0xC = PLLRAWCLK Others = Reserved Reset type: SYSRSn

3.15.10.4 SYSPLLCTL Register (Offset = 8h) [Reset = 00000000h]

SYSPLLCTL is shown in Figure 3-93 and described in Table 3-113.

Return to the Summary Table.

SYSPLL Control register
This memory mapped register requires a delay of 45 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 45 NOP instructions.

Figure 3-93. SYSPLLCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						PLLCLKEN	PLLEN
R-0-0h						R/W-0h	R/W-0h

Table 3-113. SYSPLLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	PLLCLKEN	R/W	0h	SYSPLL bypassed or included in the PLLSYSCLK path: This bit decides if the SYSPLL is bypassed when PLLSYSCLK is generated 1 = PLLSYSCLK is fed from the SYSPLL clock output. Users need to make sure that the PLL is locked before enabling this clock to the system. 0 = SYSPLL is bypassed. Clock to system is direct feed from OSCCLK Reset type: XRSn
0	PLLEN	R/W	0h	SYSPLL enabled or disabled: This bit decides if the SYSPLL is enabled or not 1 = SYSPLL is enabled 0 = SYSPLL is powered off. Clock to system is direct feed from OSCCLK Reset type: XRSn

3.15.10.5 SYSPLLMULT Register (Offset = Ah) [Reset = 00000000h]

SYSPLLMULT is shown in Figure 3-94 and described in Table 3-114.

Return to the Summary Table.

SYSPLL Multiplier register
This memory mapped register requires a delay of 45 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 45 NOP instructions.

Figure 3-94. SYSPLLMULT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						PDIV	
R-0-0h						R/W-0h	
15	14	13	12	11	10	9	8
RESERVED				RDIVCLK0			
R-0-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED	QDIV						
R-0-0h	R/W-0h						

Table 3-114. SYSPLLMULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R-0	0h	Reserved
17-16	PDIV	R/W	0h	PDIV selects the SYSPLL reference clock prescale divider. 0h = SYSPLLREF is divided by 1 1h = SYSPLLREF is divided by 2 2h = SYSPLLREF is divided by 4 3h = SYSPLLREF is divided by 8 Reset type: XRSn
15-12	RESERVED	R-0	0h	Reserved
11-8	RDIVCLK0	R/W	0h	RDIVCLK0 sets the final divider for the SYSPLLCLK0 output (Rb divider). 0h = SYSPLLCLK0 is divided by 2 1h = SYSPLLCLK0 is divided by 4 2h = SYSPLLCLK0 is divided by 6 3h = SYSPLLCLK0 is divided by 8 4h = SYSPLLCLK0 is divided by 10 ... Eh = SYSPLLCLK0 is divided by 30 Fh = SYSPLLCLK0 is divided by 32 Reset type: XRSn
7	RESERVED	R-0	0h	Reserved
6-0	QDIV	R/W	0h	QDIV selects the SYSPLL feedback path divider. 0h = Divide-by-one is not a valid QDIV option. This field should be programmed to a different value before enabling the PLL. 1h = Feedback path is divided by 2 2h = Feedback path is divided by 3 3h = Feedback path is divided by 4 ... 7Eh = Feedback path is divided by 127 7Fh = Feedback path is divided by 128 Reset type: XRSn

3.15.10.6 SYSPLLSTS Register (Offset = Ch) [Reset = 00000002h]

SYSPLLSTS is shown in Figure 3-95 and described in Table 3-115.

Return to the Summary Table.

SYSPLL Status register

Figure 3-95. SYSPLLSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					RESERVED	RESERVED	LOCKS
R-0-0h					R-0h	R-1h	R-0h

Table 3-115. SYSPLLSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	1h	Reserved
0	LOCKS	R	0h	SYSPLL Lock Status Bit: This bit indicates whether the SYSPLL is locked or not. This bit will be cleared by any write to the CLKSRCCTL1.WDHALTI bit. 0 = SYSPLL is not yet locked 1 = SYSPLL is locked Reset type: XRSn

3.15.10.7 SYSCLKDIVSEL Register (Offset = Eh) [Reset = 00000000h]

SYSCLKDIVSEL is shown in Figure 3-96 and described in Table 3-116.

Return to the Summary Table.

System Clock Divider Select register.
This memory mapped register requires a delay of 45 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 45 NOP instructions.

Figure 3-96. SYSCLKDIVSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										PLLSYSCLKDIV					
R-0-0h										R/W-0h					

Table 3-116. SYSCLKDIVSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-0	PLLSYSCLKDIV	R/W	0h	PLLSYSCLK Divide Select: This bit selects the divider setting for the PLLSYSCLK. 000000 = /1 (Default) 000001 = /2 000010 = /3 000011 = /4 000100 = /5 ..... 111111 = /64 Reset type: XRSn

3.15.10.8 XCLKOUTDIVSEL Register (Offset = 12h) [Reset = 00000003h]

XCLKOUTDIVSEL is shown in Figure 3-97 and described in Table 3-117.

Return to the Summary Table.

XCLKOUT Divider Select register
This memory mapped register requires a delay of 45 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 45 NOP instructions.

Figure 3-97. XCLKOUTDIVSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						XCLKOUTDIV	
R-0-0h						R/W-3h	

Table 3-117. XCLKOUTDIVSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1-0	XCLKOUTDIV	R/W	3h	XCLKOUT Divide Select: This bit selects the divider setting for the XCLKOUT. 00 = /1 01 = /2 10 = /4 11 = /8 (default on reset) Reset type: SYSRSn

3.15.10.9 LOSPCP Register (Offset = 14h) [Reset = 00000002h]

LOSPCP is shown in Figure 3-98 and described in Table 3-118.

Return to the Summary Table.

Low Speed Clock Source Prescalar

Figure 3-98. LOSPCP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													LSPCLKDIV		
R-0-0h													R/W-2h		

Table 3-118. LOSPCP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2-0	LSPCLKDIV	R/W	2h	These bits configure the low-speed peripheral clock (LSPCLK) rate 000,LSPCLK = / 1 001,LSPCLK = / 2 010,LSPCLK = / 4 (default on reset) 011,LSPCLK = / 6 100,LSPCLK = / 8 101,LSPCLK = / 10 110,LSPCLK = / 12 111,LSPCLK = / 14 Note: [1] This clock is used as strobe for the SCI and SPI modules. Reset type: SYSRSn

3.15.10.10 MCDCR Register (Offset = 16h) [Reset = 00000000h]

MCDCR is shown in Figure 3-99 and described in Table 3-119.

Return to the Summary Table.

Missing Clock Detect Control Register

Figure 3-99. MCDCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				OSCOFF	MCLKOFF	MCLKCLR	MCLKSTS
R-0-0h				R/W-0h	R/W-0h	R-0/W1S-0h	R-0h

Table 3-119. MCDCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	OSCOFF	R/W	0h	Oscillator Clock Disconnect from MCD Bit: 0 = OSCCLK Connected to OSCCLK Counter in MCD module 1 = OSCCLK Disconnected to OSCCLK Counter in MCD module Reset type: XRSn
2	MCLKOFF	R/W	0h	Missing Clock Detect Off Bit: 0 = Missing Clock Detect Circuit Enabled 1 = Missing Clock Detect Circuit Disabled Reset type: XRSn
1	MCLKCLR	R-0/W1S	0h	Missing Clock Clear Bit: Write 1' to this bit to clear MCLKSTS bit and reset the missing clock detect circuit.' Reset type: XRSn
0	MCLKSTS	R	0h	Missing Clock Status Bit: 0 = OSCCLK Is OK 1 = OSCCLK Detected Missing, CLOCKFAILn Generated Reset type: XRSn

3.15.10.11 X1CNT Register (Offset = 18h) [Reset = 00000000h]

X1CNT is shown in Figure 3-100 and described in Table 3-120.

Return to the Summary Table.

10-bit Counter on X1 Clock

Figure 3-100. X1CNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															CLR
R-0-0h															R-0/W1S-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					X1CNT										
R-0-0h					R-0h										

Table 3-120. X1CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R-0	0h	Reserved
16	CLR	R-0/W1S	0h	X1 Counter clear: A write of '1' to this bit field clears the X1CNT and makes it count from 0x0 again (provided X1 clock is ticking). Writes of '0' are ignore to this bit field Reset type: XRSn
15-11	RESERVED	R-0	0h	Reserved
10-0	X1CNT	R	0h	X1 Counter: - This counter increments on every X1 CLOCKs positive-edge. - Once it reaches the values of 0x7ff, it freezes - Before switching from SYSOSCBY4 to X1, application must check this counter and make sure that it has saturated. This will ensure that the Crystal connected to X1/X2 is oscillating. Reset type: XRSn

3.15.10.12 XTALCR Register (Offset = 1Ah) [Reset = 00000005h]

XTALCR is shown in Figure 3-101 and described in Table 3-121.

Return to the Summary Table.

XTAL Control Register
This memory mapped register requires a delay of 45 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 45 NOP instructions.

Figure 3-101. XTALCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					RESERVED	SE	OSCOFF
R-0-0h					R/W-1h	R/W-0h	R/W-1h

Table 3-121. XTALCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	RESERVED	R/W	1h	Reserved
1	SE	R/W	0h	Configures XTAL oscillator in single-ended or Crystal mode when XTAL oscillator is powered up(i.e. OSCOFF = 0) 0 XTAL oscillator in Crystal mode 1 XTAL oscilator in single-ended mode (through X1) Reset type: XRSn
0	OSCOFF	R/W	1h	This bit if '1', powers-down the XTAL oscillator macro and hence doesn't let X2 to be driven by the XTAL oscillator. If a crystal is connected to X1/X2, user needs to first clear this bit, wait for the oscillator to power up (using X1CNT) and then only switch the clock source to X1/X2 NOTE: Ensure no resources are using this clock source prior to disabling it. For example OSCCLKSRCSEL (SYSPLL), CANxBCLKSEL (CAN Clock), TMR2CLKSRCSEL (CPUTIMER2) and XCLKOUTSEL(XCLKOUT). Reset type: XRSn

3.15.10.13 XTALCR2 Register (Offset = 1Ch) [Reset = 00000003h]

XTALCR2 is shown in Figure 3-102 and described in Table 3-122.

Return to the Summary Table.

XTAL Control Register for pad init

Figure 3-102. XTALCR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													FEN	XOF	XIF
R-0-0h													R/W-0h	R/W-1h	R/W-1h

Table 3-122. XTALCR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved
15-3	RESERVED	R-0	0h	Reserved
2	FEN	R/W	0h	Configures XTAL oscillator pad initilisation. 0 : XOSC pads are not driven through GPIO connection. 1 : XOSC pads are driven through connected GPIO as per XIF & XOF values. This register has effect only when XOSC is OFF (no SE , no XTAL mode). If this register is set during XOSC off state (XOSCOFF=1 & SE=0) then upon change of these controls this bit gets reset and rearmed. Reset type: XRSn
1	XOF	R/W	1h	Polarity selection to initialise XO /X2 pad of the XOSC before start-up This value shall be deposited on the pad before XOSC started (XOSCOFF=1) If FEN=0 or XOSC is in XTAL or SE mode then this value will not be applied to the pad. Reset type: XRSn
0	XIF	R/W	1h	Polarity selection to initialise XI /X1 pad of the XOSC before start-up This value shall be deposited on the pad before XOSC started (XOSCOFF=1) If FEN=0 or XOSC is in XTAL or SE mode then this value will not be applied to the pad. Reset type: XRSn

3.15.10.14 CLKFAILCFG Register (Offset = 1Eh) [Reset = 00000000h]

CLKFAILCFG is shown in Figure 3-103 and described in Table 3-123.

Return to the Summary Table.

Clock Fail cause Configuration

Figure 3-103. CLKFAILCFG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	DCC0_ERROR_EN
R-0-0h						R/W-0h	R/W-0h

Table 3-123. CLKFAILCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	DCC0_ERROR_EN	R/W	0h	This field enables DCC0 Error to cause the clock-fail NMI to get asserted. 0 : DCC0 Error does not affect Clock fail NMI 1: Occurrence of DCC0 Error triggers Clock fail NMI assertion and ERROR pin assertion. Reset type: XRSn

3.15.10.15 CLKSRCSTS Register (Offset = 20h) [Reset = 00000000h]

CLKSRCSTS is shown in Figure 3-104 and described in Table 3-124.

Return to the Summary Table.

Clock Source Status

Figure 3-104. CLKSRCSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						SYSOSC_FCL_DONE	SYSOSC_ENABLED
R-0-0h						R-0h	R-0h

Table 3-124. CLKSRCSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	SYSOSC_FCL_DONE	R	0h	0 = SYSOSC FCL (Frequency Correction Loop) not done 1 = SYSOSC FCL (Frequency Correction Loop) done Note: Before enabling the SYSPLL with SYSOSCBY4 as the REFCLK, SW should wait for this bit to be set to ensure that the PLL gets an accurate reference clock. Reset type: PORESETn
0	SYSOSC_ENABLED	R	0h	0 = SYSOSC is disabled 1 = SYSOSC is enabled Reset type: PORESETn

3.15.11 CPU_SYS_REGS Registers

Table 3-125 lists the memory-mapped registers for the CPU_SYS_REGS registers. All register offset addresses not listed in Table 3-125 should be considered as reserved locations and the register contents should not be modified.

Table 3-125. CPU_SYS_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	CPUSYSLOCK	Lock bit for CPUSYS registers	EALLOW
2h	PIEVERRADDR	PIE Vector Fetch Error Address register	EALLOW
4h	SIMRESET	Simulated Reset Register	
8h	LPMCR	LPM Control Register	EALLOW
Ah	GPIOLPMSEL0	GPIO LPM Wakeup select registers	EALLOW
Ch	GPIOLPMSEL1	GPIO LPM Wakeup select registers	EALLOW
Eh	TMR2CLKCTL	Timer2 Clock Measurement functionality control register	EALLOW
10h	RESCCLR	Reset Cause Clear Register	
12h	RESC	Reset Cause register	
14h	CMPSSLPMSEL	CMPSS LPM Wakeup select registers	EALLOW
16h	CLKSTOPREQ	Peripheral Clock Stop Request Register	
18h	CLKSTOPACK	Peripheral Clock Stop Ackonwledge Register	
1Ah	USER_REG1_SYSRSn	Software Configurable registers reset by SYSRSn	
1Ch	USER_REG2_SYSRSn	Software Configurable registers reset by SYSRSn	
1Eh	USER_REG1_XRSn	Software Configurable registers reset by XRSn	
20h	USER_REG2_XRSn	Software Configurable registers reset by XRSn	
22h	USER_REG1_PORESETn	Software Configurable registers reset by PORESETn	
24h	USER_REG2_PORESETn	Software Configurable registers reset by PORESETn	
26h	USER_REG3_PORESETn	Software Configurable registers reset by PORESETn	
28h	USER_REG4_PORESETn	Software Configurable registers reset by PORESETn	
3Ch	PERCLKCR	Peripheral Clock Control	

Complex bit access types are encoded to fit into small table cells. Table 3-126 shows the codes that are used for access types in this section.

Table 3-126. CPU_SYS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 3-126. CPU_SYS_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.11.1 CPUSYSLOCK Register (Offset = 0h) [Reset = 00000000h]

CPUSYSLOCK is shown in Figure 3-105 and described in Table 3-127.

Return to the Summary Table.

Lock bit for CPUSYS registers
Notes:
[1] Any bit in this register, once set can only be cleared through a CPU1.SYSRSn. Write of 0 to any bit of this register has no effect
[2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed

Figure 3-105. CPUSYSLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						BROMPATCH	ALL
R-0-0h						R/WSonce-0h	R/WSonce-0h

Table 3-127. CPUSYSLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	BROMPATCH	R/WSonce	0h	Lock bit for BROMPATCHADDRx/DATAx registers 0: Respective register is not locked 1: Respective register is locked. Reset type: SYSRSn
0	ALL	R/WSonce	0h	Lock bit for all CPUSYS registers except BROMPATCHADDRx/DATAx 0: Respective register is not locked 1: Respective register is locked. Reset type: SYSRSn

3.15.11.2 PIEVERRADDR Register (Offset = 2h) [Reset = 003FFFFFh]

PIEVERRADDR is shown in Figure 3-106 and described in Table 3-128.

Return to the Summary Table.

PIE Vector Fetch Error Address register

Figure 3-106. PIEVERRADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										ADDR																					
R-0-0h										R/W-003FFFFFh																					

Table 3-128. PIEVERRADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	Reserved
21-0	ADDR	R/W	003FFFFFh	This register defines the address of the PIE Vector Fetch Error handler routine. Its the responsibility of user to initialize this register. If this register is not initialized, a default error handler at address 0x3fffbe will get executed. Refer to the Boot ROM section for more details on this register. Reset type: XRSn

3.15.11.3 SIMRESET Register (Offset = 4h) [Reset = 00000000h]

SIMRESET is shown in Figure 3-107 and described in Table 3-129.

Return to the Summary Table.

Simulated Reset Register
Note: This register exists only on CPU1

Figure 3-107. SIMRESET Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						XRSn	CPU1RSn
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

Table 3-129. SIMRESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write to this register succeeds only if this field is written with a value of 0xa5a5 Note: [1] Due to this KEY, only 32-bit writes will succeed (provided the KEY matches). 16-bit writes to the upper or lower half of this register will be ignored Reset type: XRSn
15-2	RESERVED	R-0	0h	Reserved
1	XRSn	R-0/W1S	0h	Writing a 1 to this field generates a XRSn like reset. Writing a 0 has no effect. Note: Writing to this pin will pull the XRSn pin low for 512 SECCLK clock cycles. Reset type: XRSn
0	CPU1RSn	R-0/W1S	0h	Writing a 1 to this field generates a reset to to CPU1. Writing a 0 has no effect. Reset type: XRSn

3.15.11.4 LPMCR Register (Offset = 8h) [Reset = 000000FCh]

LPMCR is shown in Figure 3-108 and described in Table 3-130.

Return to the Summary Table.

LPM Control Register

Figure 3-108. LPMCR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED						
R/W1S-0h	R-0-0h						
23	22	21	20	19	18	17	16
RESERVED						RESERVED	
R-0-0h						R/W-0h	
15	14	13	12	11	10	9	8
WDINTE	RESERVED						
R/W-0h	R-0-0h						
7	6	5	4	3	2	1	0
QUALSTDBY						LPM	
R/W-3Fh						R/W-0h	

Table 3-130. LPMCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W1S	0h	Reserved
30-18	RESERVED	R-0	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15	WDINTE	R/W	0h	When this bit is set to 1, it enables the watchdog interrupt signal to wake the device from STANDBY mode. Note: [1] To use this signal, the user must also enable the WDINTn signal using the WDENINT bit in the SCSR register. This signal will not wake the device from HALT mode because the clock to watchdog module is turned off Reset type: SYSRSn
14-8	RESERVED	R-0	0h	Reserved
7-2	QUALSTDBY	R/W	3Fh	Select number of OSCCLK clock cycles to qualify the selected inputs when waking the from STANDBY mode: 000000 = 2 OSCCLKs 000001 = 3 OSCCLKs ..... 111111 = 65 OSCCLKs Note: The LPMCR.QUALSTDBY register should be set to a value greater than the ratio of SECCLK/PLLSYSCLK to ensure proper wake up. Reset type: SYSRSn
1-0	LPM	R/W	0h	These bits set the low power mode for the device. Takes effect when CPU executes the IDLE instruction (when IDLE instruction is out of EXE Phase of the Pipeline) 00: IDLE Mode 01: STANDBY Mode 1x: HALT Mode Reset type: SYSRSn

3.15.11.5 GPIOLPMSEL0 Register (Offset = Ah) [Reset = 00000000h]

GPIOLPMSEL0 is shown in Figure 3-109 and described in Table 3-131.

Return to the Summary Table.

GPIO LPM Wakeup select registers
Connects the selected pin to the LPM circuit. Refer to LPM section of the TRM for the wakeup capabilities of the selected pin.

Figure 3-109. GPIOLPMSEL0 Register

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-131. GPIOLPMSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO31	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
30	GPIO30	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
29	GPIO29	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
28	GPIO28	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
27	GPIO27	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
26	GPIO26	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
25	GPIO25	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
24	GPIO24	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
23	GPIO23	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-131. GPIOLPMSEL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	GPIO22	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
21	GPIO21	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
20	GPIO20	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
19	GPIO19	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
18	GPIO18	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
17	GPIO17	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
16	GPIO16	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
15	GPIO15	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
14	GPIO14	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
13	GPIO13	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
12	GPIO12	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
11	GPIO11	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
10	GPIO10	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
9	GPIO9	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
8	GPIO8	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
7	GPIO7	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
6	GPIO6	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-131. GPIOLPMSEL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	GPIO5	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
4	GPIO4	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
3	GPIO3	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
2	GPIO2	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
1	GPIO1	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
0	GPIO0	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

3.15.11.6 GPIOLPMSEL1 Register (Offset = Ch) [Reset = 00000000h]

GPIOLPMSEL1 is shown in Figure 3-110 and described in Table 3-132.

Return to the Summary Table.

GPIO LPM Wakeup select registers
Connects the selected pin to the LPM circuit. Refer to LPM section of the TRM for the wakeup capabilities of the selected pin.

Figure 3-110. GPIOLPMSEL1 Register

31	30	29	28	27	26	25	24
GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-132. GPIOLPMSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GPIO63	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
30	GPIO62	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
29	GPIO61	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
28	GPIO60	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
27	GPIO59	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
26	GPIO58	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
25	GPIO57	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
24	GPIO56	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
23	GPIO55	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-132. GPIOLPMSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	GPIO54	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
21	GPIO53	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
20	GPIO52	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
19	GPIO51	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
18	GPIO50	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
17	GPIO49	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
16	GPIO48	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
15	GPIO47	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
14	GPIO46	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
13	GPIO45	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
12	GPIO44	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
11	GPIO43	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
10	GPIO42	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
9	GPIO41	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
8	GPIO40	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
7	GPIO39	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
6	GPIO38	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

Table 3-132. GPIOLPMSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	GPIO37	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
4	GPIO36	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
3	GPIO35	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
2	GPIO34	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
1	GPIO33	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
0	GPIO32	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

3.15.11.7 TMR2CLKCTL Register (Offset = Eh) [Reset = 00000000h]

TMR2CLKCTL is shown in Figure 3-111 and described in Table 3-133.

Return to the Summary Table.

Timer2 Clock Measurement functionality control register
This memory mapped register requires a delay of 45 SYSCLK cycles between subsequent writes to the register, otherwise a second write can be lost. This delay can be realized by adding 45 NOP instructions.

Figure 3-111. TMR2CLKCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			TMR2CLKPRESCALE			TMR2CLKSRCSEL	
R-0-0h			R/W-0h			R/W-0h	

Table 3-133. TMR2CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-3	TMR2CLKPRESCALE	R/W	0h	CPU Timer 2 Clock Pre-Scale Value: These bits select the pre-scale value for the selected clock source for CPU Timer 2: 0,0,0,/1 (default on reset) 0,0,1,/2, 0,1,0,/4 0,1,1,/8 1,0,0,/16 1,0,1,spare (defaults to /16) 1,1,0,spare (defaults to /16) 1,1,1,spare (defaults to /16) Note: [1] The CPU Timer2s Clock sync logic detects an input clock edge when configured for any clock source other than SYSCLK and generates an appropriate clock pulse to the CPU timer2. If SYSCLK is approximately the same or less then the input clock source, then the user would need to configure the pre-scale value such that SYSCLK is at least twice as fast as the pre-scaled value. Reset type: SYSRSn
2-0	TMR2CLKSRCSEL	R/W	0h	CPU Timer 2 Clock Source Select Bit: This bit selects the source for CPU Timer 2: 000 =SYSCLK Selected (default on reset, pre-scale is bypassed) 001 = WROSCBY8 010 = SYSOSCBY4 011 = XTAL 100 = PUMPOSC (from no-wrapper) 101 = FOSCCLK (Reserved) 110 = AUXPLLCLK (Reserved) 111 = reserved Reset type: SYSRSn

3.15.11.8 RESCCLR Register (Offset = 10h) [Reset = 00000000h]

RESCCLR is shown in Figure 3-112 and described in Table 3-134.

Return to the Summary Table.

Reset Cause Clear Register

Figure 3-112. RESCCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				SIMRESET_XRSn	SIMRESET_CPU1RSn	RESERVED	SCCRESETn
R-0-0h				R-0/W1S-0h	R-0/W1S-0h	R-0-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	NMIWDRSn	WDRSn	XRSn	POR
R-0-0h	R-0/W1S-0h	R-0/W1S-0h	R-0-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 3-134. RESCCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	SIMRESET_XRSn	R-0/W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
10	SIMRESET_CPU1RSn	R-0/W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
9	RESERVED	R-0	0h	Reserved
8	SCCRESETn	R-0/W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
7	RESERVED	R-0	0h	Reserved
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	RESERVED	R-0	0h	Reserved
3	NMIWDRSn	R-0/W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn

Table 3-134. RESCCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	WDRSn	R-0/W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
1	XRSn	R-0/W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn
0	POR	R-0/W1S	0h	Clear bit for corresponding status bit in RESC. Read of RESCCLR always gives 0. Writing a 1 to this bit clears the status bit in RESC to 0 Writing 0 has no effect. Reset type: SYSRSn

3.15.11.9 RESC Register (Offset = 12h) [Reset = X0000003h]

RESC is shown in Figure 3-113 and described in Table 3-135.

Return to the Summary Table.

Reset Cause register

Figure 3-113. RESC Register

31	30	29	28	27	26	25	24
DCON	XRSn_pin_status	RESERVED					
R-0h	R-Xh	R-0-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				SIMRESET_XRSn	SIMRESET_CPU1RSn	RESERVED	SCCRESETn
R-0-0h				R-0h	R-0h	R-0-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	NMIWDRSn	WDRSn	XRSn	POR
R-0-0h	R-0h	R-0h	R-0-0h	R-0h	R-0h	R-1h	R-1h

Table 3-135. RESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DCON	R	0h	Reading this bit provides the status of debugger connection to the C28x CPU. 0 : Debugger is not connected to the C28x CPU 1 : Debugger is connected to the C28x CPU Notes: [1] This bit is connected to the DCON o/p signal of the C28x CPU Reset type: N/A
30	XRSn_pin_status	R	Xh	Reading this bit provides the current status of the XRSn pin. Reset value is reflective of the pin status. Reset type: N/A
29-16	RESERVED	R-0	0h	Reserved
15-12	RESERVED	R-0	0h	Reserved
11	SIMRESET_XRSn	R	0h	If this bit is set, indicates that the device was reset by SIMRESET_XRSn Reset type: PORESETn
10	SIMRESET_CPU1RSn	R	0h	If this bit is set, indicates that the device was reset by SIMRESET_CPU1RSn Reset type: PORESETn
9	RESERVED	R-0	0h	Reserved
8	SCCRESETn	R	0h	If this bit is set, indicates that the device was reset by SCCRESETn (fired by DCSM). Reset type: PORESETn
7	RESERVED	R-0	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R-0	0h	Reserved

Table 3-135. RESC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	NMIWDRSn	R	0h	If this bit is set, indicates that the device was reset by NMIWDRSn. Note: To know the exact cause of NMI after the reset, software needs to read NMISHDFLG registers Reset type: PORESETn
2	WDRSn	R	0h	If this bit is set, indicates that the device was reset by WDRSn. Note: [1] A bit inside WD module also provides the same information. This bit is present to keep things consistent. This register is a one-stop shop for the software to know the reset cause for the C28x core. Reset type: PORESETn
1	XRSn	R	1h	If this bit is set, indicates that the device was reset by XRSn. Reset type: PORESETn
0	POR	R	1h	If this bit is set, indicates that the device was reset by PORn. Reset type: PORESETn

3.15.11.10 CMPSSLPMSEL Register (Offset = 14h) [Reset = 00000000h]

CMPSSLPMSEL is shown in Figure 3-114 and described in Table 3-136.

Return to the Summary Table.

CMPSS LPM Wakeup select registers
Connects the selected pin to the LPM circuit. Refer to LPM section of the TRM for the wakeup capabilities of the selected pin.

Figure 3-114. CMPSSLPMSEL Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CMPSS3L	CMPSS3H	CMPSS2L	CMPSS2H	CMPSS1L	CMPSS1H
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-136. CMPSSLPMSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved

Table 3-136. CMPSSLPMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	CMPSS3L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
4	CMPSS3H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
3	CMPSS2L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
2	CMPSS2H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
1	CMPSS1L	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn
0	CMPSS1H	R/W	0h	0 pin is dis-connected from LPM circuit 1 pin is connected to LPM circuit Reset type: SYSRSn

3.15.11.11 CLKSTOPREQ Register (Offset = 16h) [Reset = 00000000h]

CLKSTOPREQ is shown in Figure 3-115 and described in Table 3-137.

Return to the Summary Table.

Peripheral Clock Stop Request Register
Note: This register exists only on CPU1

Figure 3-115. CLKSTOPREQ Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED				RESERVED		RESERVED	RESERVED
R-0-0h				R-0-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		RESERVED	CAN_A	RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h		R/W-0h	R/W-0h	R-0-0h	R/W-0h	R-0-0h	R/W-0h

Table 3-137. CLKSTOPREQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write to any of the bits in this register will succeed only if a value of 0x5634 is written to the KEY field. Reset type: SYSRSn
15-12	RESERVED	R-0	0h	Reserved
11-10	RESERVED	R-0	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	CAN_A	R/W	0h	CAN_A Clock Stop Request Bit 0: If clock to CAN_A is turned off, it will be turned on, else no effect. 1: Clock stop request toCAN_A Note: Once set, this bit is cleared when clock to CAN_A is turned on as a result of a wakeup event in hardware Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R-0	0h	Reserved
0	RESERVED	R/W	0h	Reserved

3.15.11.12 CLKSTOPACK Register (Offset = 18h) [Reset = 00000000h]

CLKSTOPACK is shown in Figure 3-116 and described in Table 3-138.

Return to the Summary Table.

Peripheral Clock Stop Ackonwledge Register
Note: This register exists only on CPU1

Figure 3-116. CLKSTOPACK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED						RESERVED	RESERVED
R-0-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED		RESERVED	CAN_A	RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h		R-0h	R-0h	R-0-0h	R-0h	R-0-0h	R-0h

Table 3-138. CLKSTOPACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R-0	0h	Reserved
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R	0h	Reserved
4	CAN_A	R	0h	CAN_A Clock Stop Acknowledge Bit 0: Clock stop request not acknowledged 1: Clock stop acknowledged Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R-0	0h	Reserved
0	RESERVED	R	0h	Reserved

3.15.11.13 USER_REG1_SYSRSn Register (Offset = 1Ah) [Reset = 00000000h]

USER_REG1_SYSRSn is shown in Figure 3-117 and described in Table 3-139.

Return to the Summary Table.

Software Configurable registers reset by SYSRSn

Figure 3-117. USER_REG1_SYSRSn Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITS																															
R/W-0h																															

Table 3-139. USER_REG1_SYSRSn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by SYSRSn to be used by the application software Reset type: SYSRSn

3.15.11.14 USER_REG2_SYSRSn Register (Offset = 1Ch) [Reset = 00000000h]

USER_REG2_SYSRSn is shown in Figure 3-118 and described in Table 3-140.

Return to the Summary Table.

Software Configurable registers reset by SYSRSn

Figure 3-118. USER_REG2_SYSRSn Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITS																															
R/W-0h																															

Table 3-140. USER_REG2_SYSRSn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by SYSRSn to be used by the application software Reset type: SYSRSn

3.15.11.15 USER_REG1_XRSn Register (Offset = 1Eh) [Reset = 00000000h]

USER_REG1_XRSn is shown in Figure 3-119 and described in Table 3-141.

Return to the Summary Table.

Software Configurable registers reset by XRSn

Figure 3-119. USER_REG1_XRSn Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITS																															
R/W-0h																															

Table 3-141. USER_REG1_XRSn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by XRSn to be used by the application software Reset type: XRSn

3.15.11.16 USER_REG2_XRSn Register (Offset = 20h) [Reset = 00000000h]

USER_REG2_XRSn is shown in Figure 3-120 and described in Table 3-142.

Return to the Summary Table.

Software Configurable registers reset by XRSn

Figure 3-120. USER_REG2_XRSn Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITS																															
R/W-0h																															

Table 3-142. USER_REG2_XRSn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by XRSn to be used by the application software Reset type: XRSn

3.15.11.17 USER_REG1_PORESETn Register (Offset = 22h) [Reset = 00000000h]

USER_REG1_PORESETn is shown in Figure 3-121 and described in Table 3-143.

Return to the Summary Table.

Software Configurable registers reset by PORESETn

Figure 3-121. USER_REG1_PORESETn Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITS																															
R/W-0h																															

Table 3-143. USER_REG1_PORESETn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by PORESETn to be used by the application software Reset type: PORESETn

3.15.11.18 USER_REG2_PORESETn Register (Offset = 24h) [Reset = 00000000h]

USER_REG2_PORESETn is shown in Figure 3-122 and described in Table 3-144.

Return to the Summary Table.

Software Configurable registers reset by PORESETn

Figure 3-122. USER_REG2_PORESETn Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITS																															
R/W-0h																															

Table 3-144. USER_REG2_PORESETn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by PORESETn to be used by the application software Reset type: PORESETn

3.15.11.19 USER_REG3_PORESETn Register (Offset = 26h) [Reset = 00000000h]

USER_REG3_PORESETn is shown in Figure 3-123 and described in Table 3-145.

Return to the Summary Table.

Software Configurable registers reset by PORESETn

Figure 3-123. USER_REG3_PORESETn Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITS																															
R/W-0h																															

Table 3-145. USER_REG3_PORESETn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by PORESETn to be used by the application software Reset type: PORESETn

3.15.11.20 USER_REG4_PORESETn Register (Offset = 28h) [Reset = 00000000h]

USER_REG4_PORESETn is shown in Figure 3-124 and described in Table 3-146.

Return to the Summary Table.

Software Configurable registers reset by PORESETn

Figure 3-124. USER_REG4_PORESETn Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITS																															
R/W-0h																															

Table 3-146. USER_REG4_PORESETn Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITS	R/W	0h	R/W bits reset by PORESETn to be used by the application software Reset type: PORESETn

3.15.11.21 PERCLKCR Register (Offset = 3Ch) [Reset = 00000000h]

PERCLKCR is shown in Figure 3-125 and described in Table 3-147.

Return to the Summary Table.

Peripheral Clock Control

Figure 3-125. PERCLKCR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	TBCLKSYNC
R-0-0h						R/W-0h	R/W-0h

Table 3-147. PERCLKCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	TBCLKSYNC	R/W	0h	PWM Time Base Clock sync: When set PWM time bases of all the PWM modules belonging to the same CPU-Subsystem (as partitioned using their CPUSEL bits) start counting Reset type: SYSRSn

3.15.12 SYS_STATUS_REGS Registers

Table 3-148 lists the memory-mapped registers for the SYS_STATUS_REGS registers. All register offset addresses not listed in Table 3-148 should be considered as reserved locations and the register contents should not be modified.

Table 3-148. SYS_STATUS_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	SYS_ERR_INT_FLG	Status of interrupts due to multiple different errors in the system.	
2h	SYS_ERR_INT_CLR	SYS_ERR_INT_FLG clear register	
4h	SYS_ERR_INT_SET	SYS_ERR_INT_FLG set register	EALLOW
6h	SYS_ERR_MASK	SYS_ERR_MASK register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 3-149 shows the codes that are used for access types in this section.

Table 3-149. SYS_STATUS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.12.1 SYS_ERR_INT_FLG Register (Offset = 0h) [Reset = 00000000h]

SYS_ERR_INT_FLG is shown in Figure 3-126 and described in Table 3-150.

Return to the Summary Table.

Status of interrupts due to multiple different errors in the system.

Figure 3-126. SYS_ERR_INT_FLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	FPU_OFLOW	FPU_UFLOW
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CORRECTABLE_ERR	RESERVED	GINT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 3-150. SYS_ERR_INT_FLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	FPU_OFLOW	R	0h	0: FPU_OFLOW has not fired an interrupt. 1: FPU_OFLOW has fired an interrupt Reset type: SYSRSn
16	FPU_UFLOW	R	0h	0: FPU_UFLOW has not fired an interrupt. 1: FPU_UFLOW has fired an interrupt Reset type: SYSRSn
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved

Table 3-150. SYS_ERR_INT_FLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CORRECTABLE_ERR	R	0h	0: Number of correctable errors detected has not exceeded the set threshold for flash/RAM. 1:Number of correctable errors detected has exceeded the set threshold for flash/RAM. Reset type: SYSRSn
1	RESERVED	R	0h	Reserved
0	GINT	R	0h	Global Interrupt flag: 0: On any of the flags of SYS_ERR_INT_FLG register being set, SYS_ERR_INT is pulsed and GINT flag would be set 1: No further interrupts would be fired until GINT flag is cleared Reset type: SYSRSn

3.15.12.2 SYS_ERR_INT_CLR Register (Offset = 2h) [Reset = 00000000h]

SYS_ERR_INT_CLR is shown in Figure 3-127 and described in Table 3-151.

Return to the Summary Table.

SYS_ERR_INT_FLG clear register

Figure 3-127. SYS_ERR_INT_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	FPU_OFLOW	FPU_UFLOW
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CORRECTABLE_ERR	RESERVED	GINT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 3-151. SYS_ERR_INT_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	RESERVED	R-0/W1S	0h	Reserved
18	RESERVED	R-0/W1S	0h	Reserved
17	FPU_OFLOW	R-0/W1S	0h	0: No effect 1: FPU_OFLOW flag of SYS_ERR_INT_FLG reister will be cleared. Reset type: SYSRSn
16	FPU_UFLOW	R-0/W1S	0h	0: No effect 1: FPU_UFLOW flag of SYS_ERR_INT_FLG reister will be cleared. Reset type: SYSRSn
15	RESERVED	R-0/W1S	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13	RESERVED	R-0/W1S	0h	Reserved
12	RESERVED	R-0/W1S	0h	Reserved
11	RESERVED	R-0/W1S	0h	Reserved
10	RESERVED	R-0/W1S	0h	Reserved
9	RESERVED	R-0/W1S	0h	Reserved
8	RESERVED	R-0/W1S	0h	Reserved
7	RESERVED	R-0/W1S	0h	Reserved
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	RESERVED	R-0/W1S	0h	Reserved
3	RESERVED	R-0/W1S	0h	Reserved
2	CORRECTABLE_ERR	R-0/W1S	0h	0: No effect 1: CORRECTABLE_ERR flag of SYS_ERR_INT_FLG reister will be cleared. Reset type: SYSRSn

Table 3-151. SYS_ERR_INT_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RESERVED	R-0/W1S	0h	Reserved
0	GINT	R-0/W1S	0h	0: No effect 1: GINT flag of SYS_ERR_INT_FLG reister will be cleared. Reset type: SYSRSn

3.15.12.3 SYS_ERR_INT_SET Register (Offset = 4h) [Reset = 00000000h]

SYS_ERR_INT_SET is shown in Figure 3-128 and described in Table 3-152.

Return to the Summary Table.

SYS_ERR_INT_FLG set register

Figure 3-128. SYS_ERR_INT_SET Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	FPU_OFLOW	FPU_UFLOW
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CORRECTABLE_ERR	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 3-152. SYS_ERR_INT_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	A value of 0xa5 to this field would enable write to the other bit fields of this register. Any other value written to KEY field would block the write to the other fields of this register. Note: Only a 32 bit write to this register will succeed in updating the fields of this rigister, provided the correct value written to the KEY field simultaneously Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19	RESERVED	R-0/W1S	0h	Reserved
18	RESERVED	R-0/W1S	0h	Reserved
17	FPU_OFLOW	R-0/W1S	0h	0: No effect 1: FPU_OFLOW flag of SYS_ERR_INT_FLG reister will be set. Reset type: SYSRSn
16	FPU_UFLOW	R-0/W1S	0h	0: No effect 1: FPU_UFLOW flag of SYS_ERR_INT_FLG reister will be set. Reset type: SYSRSn
15	RESERVED	R-0/W1S	0h	Reserved
14	RESERVED	R-0/W1S	0h	Reserved
13	RESERVED	R-0/W1S	0h	Reserved
12	RESERVED	R-0/W1S	0h	Reserved
11	RESERVED	R-0/W1S	0h	Reserved
10	RESERVED	R-0/W1S	0h	Reserved
9	RESERVED	R-0/W1S	0h	Reserved
8	RESERVED	R-0/W1S	0h	Reserved
7	RESERVED	R-0/W1S	0h	Reserved
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved

Table 3-152. SYS_ERR_INT_SET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	RESERVED	R-0/W1S	0h	Reserved
3	RESERVED	R-0/W1S	0h	Reserved
2	CORRECTABLE_ERR	R-0/W1S	0h	0: No effect 1: CORRECTABLE_ERR flag of SYS_ERR_INT_FLG reister will be set. Reset type: SYSRSn
1	RESERVED	R-0/W1S	0h	Reserved
0	RESERVED	R	0h	Reserved

3.15.12.4 SYS_ERR_MASK Register (Offset = 6h) [Reset = 00000000h]

SYS_ERR_MASK is shown in Figure 3-129 and described in Table 3-153.

Return to the Summary Table.

SYS_ERR_MASK register

Figure 3-129. SYS_ERR_MASK Register

31	30	29	28	27	26	25	24
KEY							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	FPU_OFLOW	FPU_UFLOW
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CORRECTABLE_ERR	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 3-153. SYS_ERR_MASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R/W	0h	A value of 0xa5 to this field would enable write to the other bit fields of this register. Any other value written to KEY field would block the write to the other fields of this register. Note: Only a 32 bit write to this register will succeed in updating the fields of this rigister, provided the correct value written to the KEY field simultaneously Reset type: SYSRSn
23-20	RESERVED	R	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	FPU_OFLOW	R/W	0h	0: FPU_OFLOW flag of SYS_ERR_INT_FLG reister will be set on a hardware event. 1: FPU_OFLOW flag of SYS_ERR_INT_FLG reister will not be set on a hardware event. Reset type: SYSRSn
16	FPU_UFLOW	R/W	0h	0: FPU_UFLOW flag of SYS_ERR_INT_FLG reister will be set on a hardware event. 1: FPU_UFLOW flag of SYS_ERR_INT_FLG reister will not be set on a hardware event. Reset type: SYSRSn
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved

Table 3-153. SYS_ERR_MASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	CORRECTABLE_ERR	R/W	0h	0: CORRECTABLE_ERR flag of SYS_ERR_INT_FLG reister will be set on a hardware event. 1: CORRECTABLE_ERR flag of SYS_ERR_INT_FLG reister will not be set on a hardware event. Reset type: SYSRSn
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R	0h	Reserved

3.15.13 MEM_CFG_REGS Registers

Table 3-154 lists the memory-mapped registers for the MEM_CFG_REGS registers. All register offset addresses not listed in Table 3-154 should be considered as reserved locations and the register contents should not be modified.

Table 3-154. MEM_CFG_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	DxLOCK	Dedicated RAM config Lock Register	EALLOW
2h	DxCOMMIT	Dedicated RAM config Lock Commit Register	EALLOW
4h	DxTEST	Dedicated RAM TEST Register	
6h	DxINIT	Dedicated RAM Init Register	EALLOW
8h	DxINITDONE	Dedicated RAM InitDone Status Register	
Ah	DxRAMTEST_LOCK	Dedicated RAM TEST Lock Register	
20h	GSxLOCK	Global Shared RAM Config Lock Register	EALLOW
22h	GSxCOMMIT	Global Shared RAM conffig Lock Commit Register	EALLOW
24h	GSxTEST	Global Shared RAM TEST Register	
26h	GSxINIT	Global Shared RAM Init Register	EALLOW
28h	GSxINITDONE	Global Shared RAM InitDone Status Register	
2Ah	GSxRAMTEST_LOCK	Global Shared RAM TEST Lock Register	
30h	ROM_LOCK	Rom configuration lock register	
32h	ROM_TEST	ROM TEST Register	
34h	ROM_FORCE_ERROR	ROM Force Error register	

Complex bit access types are encoded to fit into small table cells. Table 3-155 shows the codes that are used for access types in this section.

Table 3-155. MEM_CFG_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.13.1 DxLOCK Register (Offset = 0h) [Reset = 00000000h]

DxLOCK is shown in Figure 3-130 and described in Table 3-156.

Return to the Summary Table.

Dedicated RAM config Lock Register

Figure 3-130. DxLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					LOCK_PIEVECT	LOCK_M1	LOCK_M0
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 3-156. DxLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	LOCK_PIEVECT	R/W	0h	Locks the write to access protection, controller select, initialization control and test register fields for PIEVECT RAM: 0: Write to INIT fields are allowed. 1: Write to INIT fields are blocked. Reset type: SYSRSn
1	LOCK_M1	R/W	0h	Locks the write to access protection, controller select, initialization control and test register fields for M1 RAM: 0: Write to INIT fields are allowed. 1: Write to INIT fields are blocked. Reset type: SYSRSn
0	LOCK_M0	R/W	0h	Locks the write to access protection, controller select, initialization control and test register fields for M0 RAM: 0: Write to INIT fields are allowed. 1: Write to INIT fields are blocked. Reset type: SYSRSn

3.15.13.2 DxCOMMIT Register (Offset = 2h) [Reset = 00000000h]

DxCOMMIT is shown in Figure 3-131 and described in Table 3-157.

Return to the Summary Table.

Dedicated RAM config Lock Commit Register

Figure 3-131. DxCOMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					COMMIT_PIEVECT	COMMIT_M1	COMMIT_M0
R-0-0h					R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 3-157. DxCOMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	COMMIT_PIEVECT	R/WSonce	0h	Permanently Locks the write to access protection, controller select, initialization control and test register fields for PIEVECT RAM: 0: Write to INIT fields are allowed based on value of lock field in DxLOCK register. 1: Write to INIT fields are permanently blocked. Reset type: SYSRSn
1	COMMIT_M1	R/WSonce	0h	Permanently Locks the write to access protection, controller select, initialization control and test register fields for M1 RAM: 0: Write to INIT fields are allowed based on value of lock field in DxLOCK register. 1: Write to INIT fields are permanently blocked. Reset type: SYSRSn
0	COMMIT_M0	R/WSonce	0h	Permanently Locks the write to access protection, controller select, initialization control and test register fields for M0 RAM: 0: Write to INIT fields are allowed based on value of lock field in DxLOCK register. 1: Write to INIT fields are permanently blocked. Reset type: SYSRSn

3.15.13.3 DxTEST Register (Offset = 4h) [Reset = 00000000h]

DxTEST is shown in Figure 3-132 and described in Table 3-158.

Return to the Summary Table.

Dedicated RAM TEST Register

Figure 3-132. DxTEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		TEST_PIEVECT		TEST_M1		TEST_M0	
R-0-0h		R/W-0h		R/W-0h		R/W-0h	

Table 3-158. DxTEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-4	TEST_PIEVECT	R/W	0h	Selects the defferent modes for PIEVECT RAM: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to Parity bits. 10: Writes are allowed to Parity bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn
3-2	TEST_M1	R/W	0h	Selects the defferent modes for M1 RAM: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to Parity bits. 10: Writes are allowed to Parity bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn
1-0	TEST_M0	R/W	0h	Selects the defferent modes for M0 RAM: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to Parity bits. 10: Writes are allowed to Parity bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn

3.15.13.4 DxINIT Register (Offset = 6h) [Reset = 00000000h]

DxINIT is shown in Figure 3-133 and described in Table 3-159.

Return to the Summary Table.

Dedicated RAM Init Register

Figure 3-133. DxINIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					INIT_PIEVECT	INIT_M1	INIT_M0
R-0-0h					R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 3-159. DxINIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	INIT_PIEVECT	R-0/W1S	0h	RAM Initialization control for PIEVECT RAM: 0: None. 1: Start RAM Initialization. Reset type: SYSRSn
1	INIT_M1	R-0/W1S	0h	RAM Initialization control for M1 RAM: 0: None. 1: Start RAM Initialization. Reset type: SYSRSn
0	INIT_M0	R-0/W1S	0h	RAM Initialization control for M0 RAM: 0: None. 1: Start RAM Initialization. Reset type: SYSRSn

3.15.13.5 DxINITDONE Register (Offset = 8h) [Reset = 00000000h]

DxINITDONE is shown in Figure 3-134 and described in Table 3-160.

Return to the Summary Table.

Dedicated RAM InitDone Status Register

Figure 3-134. DxINITDONE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					INITDONE_PIEVECT	INITDONE_M1	INITDONE_M0
R-0-0h					R-0h	R-0h	R-0h

Table 3-160. DxINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	INITDONE_PIEVECT	R	0h	RAM Initialization status for PIEVECT RAM: 0: RAM Initialization has completed. 1: RAM Initialization has completed. Reset type: SYSRSn
1	INITDONE_M1	R	0h	RAM Initialization status for M1 RAM: 0: RAM Initialization is not done. 1: RAM Initialization has completed. Reset type: SYSRSn
0	INITDONE_M0	R	0h	RAM Initialization status for M0 RAM: 0: RAM Initialization is not done. 1: RAM Initialization is done. Reset type: SYSRSn

3.15.13.6 DxRAMTEST_LOCK Register (Offset = Ah) [Reset = 00000000h]

DxRAMTEST_LOCK is shown in Figure 3-135 and described in Table 3-161.

Return to the Summary Table.

Dedicated RAM TEST Lock Register

Figure 3-135. DxRAMTEST_LOCK Register

31	30	29	28	27	26	25	24
Key							
R-0/W-0h							
23	22	21	20	19	18	17	16
Key							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					PIEVECT_TEST	M1_TEST	M0_TEST
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 3-161. DxRAMTEST_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Key	R-0/W	0h	A value of 0xa5a5 to this field is simultaneously required for the writes to the rest of the fields of this register to succeed, Reset type: SYSRSn
15-3	RESERVED	R-0	0h	Reserved
2	PIEVECT_TEST	R/W	0h	0: Allows writes to RAM_TEST_PIEVECT field. 1: Blocks writes to RAM_TEST_PIEVECT field. Reset type: SYSRSn
1	M1_TEST	R/W	0h	0: Allows writes to RAM_TEST_M1 field. 1: Blocks writes to RAM_TEST_M1 field. Reset type: SYSRSn
0	M0_TEST	R/W	0h	0: Allows writes to RAM_TEST M0 field. 1: Blocks writes to RAM_TEST M0 field. Reset type: SYSRSn

3.15.13.7 GSxLOCK Register (Offset = 20h) [Reset = 00000000h]

GSxLOCK is shown in Figure 3-136 and described in Table 3-162.

Return to the Summary Table.

Global Shared RAM Config Lock Register

Figure 3-136. GSxLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	LOCK_GS0
R-0-0h						R/W-0h	R/W-0h

Table 3-162. GSxLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	LOCK_GS0	R/W	0h	Locks the write to access protection, controller select, initialization control and test register fields for GS0 RAM: 0: Write to INIT fields are allowed. 1: Write to INIT fields are blocked. Reset type: SYSRSn

3.15.13.8 GSxCOMMIT Register (Offset = 22h) [Reset = 00000000h]

GSxCOMMIT is shown in Figure 3-137 and described in Table 3-163.

Return to the Summary Table.

Global Shared RAM conffig Lock Commit Register

Figure 3-137. GSxCOMMIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	COMMIT_GS0
R-0-0h						R/WSonce-0h	R/WSonce-0h

Table 3-163. GSxCOMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/WSonce	0h	Reserved
0	COMMIT_GS0	R/WSonce	0h	Permanently Locks the write to access protection, controller select, initialization control and test register fields for GS0 RAM: 0: Write to INIT fields are allowed based on value of lock field in DxLOCK register. 1: Write to INIT fields are permanently blocked. Reset type: SYSRSn

3.15.13.9 GSxTEST Register (Offset = 24h) [Reset = 00000000h]

GSxTEST is shown in Figure 3-138 and described in Table 3-164.

Return to the Summary Table.

Global Shared RAM TEST Register

Figure 3-138. GSxTEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED		TEST_GS0	
R-0-0h				R/W-0h		R/W-0h	

Table 3-164. GSxTEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3-2	RESERVED	R/W	0h	Reserved
1-0	TEST_GS0	R/W	0h	Selects the defferent modes for GS0 RAM: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to ECC bits. 10: Writes are allowed to ECC bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn

3.15.13.10 GSxINIT Register (Offset = 26h) [Reset = 00000000h]

GSxINIT is shown in Figure 3-139 and described in Table 3-165.

Return to the Summary Table.

Global Shared RAM Init Register

Figure 3-139. GSxINIT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	INIT_GS0
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

Table 3-165. GSxINIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R-0/W1S	0h	Reserved
0	INIT_GS0	R-0/W1S	0h	RAM Initialization control for GS0 RAM: 0: None. 1: Start RAM Initialization. Reset type: SYSRSn

3.15.13.11 GSxINITDONE Register (Offset = 28h) [Reset = 00000000h]

GSxINITDONE is shown in Figure 3-140 and described in Table 3-166.

Return to the Summary Table.

Global Shared RAM InitDone Status Register

Figure 3-140. GSxINITDONE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	INITDONE_GS0
R-0-0h						R-0h	R-0h

Table 3-166. GSxINITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R	0h	Reserved
0	INITDONE_GS0	R	0h	RAM Initialization status for GS0 RAM: 0: RAM Initialization is not done. 1: RAM Initialization is done. Reset type: SYSRSn

3.15.13.12 GSxRAMTEST_LOCK Register (Offset = 2Ah) [Reset = 00000000h]

GSxRAMTEST_LOCK is shown in Figure 3-141 and described in Table 3-167.

Return to the Summary Table.

Global Shared RAM TEST Lock Register

Figure 3-141. GSxRAMTEST_LOCK Register

31	30	29	28	27	26	25	24
Key							
R-0/W-0h							
23	22	21	20	19	18	17	16
Key							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	GS0_TEST
R-0-0h						R/W-0h	R/W-0h

Table 3-167. GSxRAMTEST_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Key	R-0/W	0h	A value of 0xa5a5 to this field is simultaneously required for the writes to the rest of the fields of this register to succeed, Reset type: SYSRSn
15-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	GS0_TEST	R/W	0h	0: Allows writes toRAM_TEST_GS0 field. 1: Blocks writes to RAM_TEST_GS0 field. Reset type: SYSRSn

3.15.13.13 ROM_LOCK Register (Offset = 30h) [Reset = 00000000h]

ROM_LOCK is shown in Figure 3-142 and described in Table 3-168.

Return to the Summary Table.

Rom configuration lock register

Figure 3-142. ROM_LOCK Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							Lock_BOOTROM_SECUREROM
R-0-0h							R/W-0h

Table 3-168. ROM_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	A value of 0xa5a5 to this field is simultaneously required for the writes to the rest of the fields of this register to succeed, Reset type: SYSRSn
15-1	RESERVED	R-0	0h	Reserved
0	Lock_BOOTROM_SECUREROM	R/W	0h	Locks write access to test control fields (TEST and FORCE_ERROR) of BOOTROM 0: Write access allowed 1: Write access blocked Reset type: SYSRSn

3.15.13.14 ROM_TEST Register (Offset = 32h) [Reset = 00000000h]

ROM_TEST is shown in Figure 3-143 and described in Table 3-169.

Return to the Summary Table.

ROM TEST Register

Figure 3-143. ROM_TEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						TEST_BOOTROM_SECUREROM	
R-0-0h						R/W-0h	

Table 3-169. ROM_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1-0	TEST_BOOTROM_SECUREROM	R/W	0h	Selects the different modes for BOOTROM nad SECURE ROM: 00: Functional Mode. 01: same as '00' but Parity check on data read is disabled (for debug) 10: Parity Bits are visible on memory map (for debug) 11: Same as '00' but NMI is not generated on errors, used for diagnostics. (for diagnostics) Reset type: SYSRSn

3.15.13.15 ROM_FORCE_ERROR Register (Offset = 34h) [Reset = 00000000h]

ROM_FORCE_ERROR is shown in Figure 3-144 and described in Table 3-170.

Return to the Summary Table.

ROM Force Error register

Figure 3-144. ROM_FORCE_ERROR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							FORCE_BOOTROM_SECUREROM_ERROR
R-0-0h							R/W-0h

Table 3-170. ROM_FORCE_ERROR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	FORCE_BOOTROM_SECUREROM_ERROR	R/W	0h	Force parity error by feeding inverted Parity bit to Parity checking logic. Reset type: SYSRSn

3.15.14 MEMORY_ERROR_REGS Registers

Table 3-171 lists the memory-mapped registers for the MEMORY_ERROR_REGS registers. All register offset addresses not listed in Table 3-171 should be considered as reserved locations and the register contents should not be modified.

Table 3-171. MEMORY_ERROR_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	UCERRFLG	Uncorrectable Error Flag Register	
2h	UCERRSET	Uncorrectable Error Flag Set Register	EALLOW
4h	UCERRCLR	Uncorrectable Error Flag Clear Register	EALLOW
6h	UCCPUREADDR	Uncorrectable CPU Error Address	
8h	UCDMAREADDR	Uncorrectable DMA Error Address	
10h	FLUCERRSTATUS	Flash read uncorrectable ecc err status	
12h	FLCERRSTATUS	Flash read correctable ecc err status	
20h	CERRFLG	Correctable Error Flag Register	
22h	CERRSET	Correctable Error Flag Set Register	
24h	CERRCLR	Correctable Error Flag Clear Register	
26h	CCPUREADDR	Correctable CPU Error Address	
28h	CDMAREADDR	Correctable DMA Error Address	
30h	CERRCNT	Correctable Error Count Register	
32h	CERRTHRES	Correctable Error Threshold Value Register	EALLOW
34h	CEINTFLG	Correctable Error Interrupt Flag Status Register	
36h	CEINTCLR	Correctable Error Interrupt Flag Clear Register	EALLOW
38h	CEINTSET	Correctable Error Interrupt Flag Set Register	EALLOW
3Ah	CEINTEN	Correctable Error Interrupt Enable Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 3-172 shows the codes that are used for access types in this section.

Table 3-172. MEMORY_ERROR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.

Table 3-172. MEMORY_ERROR_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.14.1 UCERRFLG Register (Offset = 0h) [Reset = 00000000h]

UCERRFLG is shown in Figure 3-145 and described in Table 3-173.

Return to the Summary Table.

Uncorrectable Error Flag Register

Figure 3-145. UCERRFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	DMA_READ_ERR	CPU_READ_ERR
R-0-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 3-173. UCERRFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	DMA_READ_ERR	R	0h	DMA Uncorrectable Read Error Flag 0: No Error. 1: Uncorrectable error occurred during DMA read. Reset type: SYSRSn
0	CPU_READ_ERR	R	0h	CPU Uncorrectable Read Error Flag 0: No Error. 1: Uncorrectable error occurred during CPU read. Reset type: SYSRSn

3.15.14.2 UCERRSET Register (Offset = 2h) [Reset = 00000000h]

UCERRSET is shown in Figure 3-146 and described in Table 3-174.

Return to the Summary Table.

Uncorrectable Error Flag Set Register

Figure 3-146. UCERRSET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	DMA_READ_ERR	CPU_READ_ERR
R-0-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 3-174. UCERRSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	RESERVED	R-0/W1S	0h	Reserved
3	RESERVED	R-0/W1S	0h	Reserved
2	RESERVED	R-0/W1S	0h	Reserved
1	DMA_READ_ERR	R-0/W1S	0h	DMA Uncorrectable Read Error Flag Set 0: No Error. 1: DMA Read Error Flag in UCERRFLG register will be set and interrupt will be generated if enabled Reset type: SYSRSn
0	CPU_READ_ERR	R-0/W1S	0h	CPU Uncorrectable Read Error Flag Set 0: No Error. 1: CPU Read Error Flag in UCERRFLG register will be set and interrupt will be generated if enabled Reset type: SYSRSn

3.15.14.3 UCERRCLR Register (Offset = 4h) [Reset = 00000000h]

UCERRCLR is shown in Figure 3-147 and described in Table 3-175.

Return to the Summary Table.

Uncorrectable Error Flag Clear Register

Figure 3-147. UCERRCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	DMA_READ_ERR	CPU_READ_ERR
R-0-0h		R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 3-175. UCERRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R-0/W1C	0h	Reserved
4	RESERVED	R-0/W1C	0h	Reserved
3	RESERVED	R-0/W1C	0h	Reserved
2	RESERVED	R-0/W1C	0h	Reserved
1	DMA_READ_ERR	R-0/W1C	0h	DMA Uncorrectable Read Error Flag clear 0: No Error. 1:DMA Read error Flag in UCERRFLG register will be cleared Reset type: SYSRSn
0	CPU_READ_ERR	R-0/W1C	0h	CPU Uncorrectable Read Error Flag clear 0: No Error. 1: CPU Read error Flag in UCERRFLG register will be cleared Reset type: SYSRSn

3.15.14.4 UCCPUREADDR Register (Offset = 6h) [Reset = 00000000h]

UCCPUREADDR is shown in Figure 3-148 and described in Table 3-176.

Return to the Summary Table.

Uncorrectable CPU Error Address

Figure 3-148. UCCPUREADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UCCPUREADDR																															
R-0h																															

Table 3-176. UCCPUREADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UCCPUREADDR	R	0h	This register captures the address location for which CPU read/fetch access resulted in uncorrectable ECC/Parity error. Reset type: SYSRSn

3.15.14.5 UCDMAREADDR Register (Offset = 8h) [Reset = 00000000h]

UCDMAREADDR is shown in Figure 3-149 and described in Table 3-177.

Return to the Summary Table.

Uncorrectable DMA Error Address

Figure 3-149. UCDMAREADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UCDMAREADDR																															
R-0h																															

Table 3-177. UCDMAREADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UCDMAREADDR	R	0h	This register captures the address location for which DMA read/fetch access resulted in uncorrectable ECC/Parity error. Reset type: SYSRSn

3.15.14.6 FLUCERRSTATUS Register (Offset = 10h) [Reset = 00000000h]

FLUCERRSTATUS is shown in Figure 3-150 and described in Table 3-178.

Return to the Summary Table.

Flash read uncorrectable ecc err status

Figure 3-150. FLUCERRSTATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						DIAG_H_FAIL	UNC_ERR_H
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED						DIAG_L_FAIL	UNC_ERR_L
R-0h						R-0h	R-0h

Table 3-178. FLUCERRSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9	DIAG_H_FAIL	R	0h	Status of ECC diagnosic logic . 0 : Status of diagnostic error if there is no single/mutli bit error on the output of ECC check. 1 : Status of diagnostic error if there is single/mutli bit error on the output of ECC check. Note : The field gets updated along with UNC_ERR_H This flag is cleared by writing a 1 to UCERRCLR.CPURDERR flag UCERRSET.CPURDERR has no effect on this flag. Reset type: SYSRSn
8	UNC_ERR_H	R	0h	Uncorrectable error. A value of 1 indicates that an un-correctable error occurred in upper 64bits of a 128-bit aligned address. Note : This flag is cleared by writing a 1 to UCERRCLR.CPURDERR flag UCERRSET.CPURDERR has no effect on this flag. Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1	DIAG_L_FAIL	R	0h	Status of ECC diagnosic logic . 0 : Status of diagnostic error if there is no single/mutli bit error on the output of ECC check. 1 : Status of diagnostic error if there is single/mutli bit error on the output of ECC check. Note : The field gets updated along with UNC_ERR_L This flag is cleared by writing a 1 to UCERRCLR.CPURDERR flag UCERRSET.CPURDERR has no effect on this flag. Reset type: SYSRSn

Table 3-178. FLUCERRSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	UNC_ERR_L	R	0h	Uncorrectable error. A value of 1 indicates that an un-correctable error occurred in lower 64bits of a 128-bit aligned address. Note : This flag is cleared by writing a 1 to UCERRCLR.CPURDERR flag UCERRSET.CPURDERR has no effect on this flag. Reset type: SYSRSn

3.15.14.7 FLCERRSTATUS Register (Offset = 12h) [Reset = 00000000h]

FLCERRSTATUS is shown in Figure 3-151 and described in Table 3-179.

Return to the Summary Table.

Flash read correctable ecc err status

Figure 3-151. FLCERRSTATUS Register

31	30	29	28	27	26	25	24
RESERVED		ERR_TYPE_H	ERR_POS_H				
R-0h		R-0h	R-0h				
23	22	21	20	19	18	17	16
ERR_POS_H	ERR_TYPE_L	ERR_POS_L					
R-0h	R-0h	R-0h					
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	FAIL_1_H	FAIL_0_H	RESERVED	FAIL_1_L	FAIL_0_L
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 3-179. FLCERRSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	Reserved
29	ERR_TYPE_H	R	0h	Error type 0 Indicates that a single bit error occured in upper 64 data bits of a 128-bit aligned address. 1 Indicates that a single bit error occured in ECC check bits of upper 64bits of a 128-bit aligned address. Note : This field is cleared by writing a 1 to CERRCLR.CPURDERR flag CERRSET.CPURDERR has no effect on this field. Reset type: SYSRSn
28-23	ERR_POS_H	R	0h	Error position. Bit position of the single bit error in upper 64bits of a 128-bit aligned address. The position is interpreted depending on whether the ERR_TYPE bit indicates a check bit or a data bit. If ERR_TYPE indicates a check bit error, the error position could range from 0 to 7, else it could range from 0 to 63. Note : This field is cleared by writing a 1 to CERRCLR.CPURDERR flag CERRSET.CPURDERR has no effect on this field. Reset type: SYSRSn
22	ERR_TYPE_L	R	0h	Error type 0 Indicates that a single bit error occured in lower 64 data bits of a 128-bit aligned address. 1 Indicates that a single bit error occured in ECC check bits of lower 64bits of a 128-bit aligned address. Note : This field is cleared by writing a 1 to CERRCLR.CPURDERR flag CERRSET.CPURDERR has no effect on this field. Reset type: SYSRSn

Table 3-179. FLCERRSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21-16	ERR_POS_L	R	0h	Error position. Bit position of the single bit error in lower 64bits of a 128-bit aligned address. The position is interpreted depending on whether the ERR_TYPE bit indicates a check bit or a data bit. If ERR_TYPE indicates a check bit error, the error position could range from 0 to 7, else it could range from 0 to 63. Note : This field is cleared by writing a 1 to CERRCLR.CPURDERR flag CERRSET.CPURDERR has no effect on this field. Reset type: SYSRSn
15-6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	FAIL_1_H	R	0h	Fail on 1. 0 Fail on 1 single bit error did not occur in upper 64bits of a 128-bit aligned address. 1 A value of 1 would indicate that a single bit error occurred in upper 64bits of a 128-bit aligned address and the corrected value was 1. Note : This flag is cleared by writing a 1 to CERRCLR.CPURDERR flag CERRSET.CPURDERR has no effect on this flag. Reset type: SYSRSn
3	FAIL_0_H	R	0h	Fail on 0. 0 Fail on 0 single bit error did not occur in upper 64bits of a 128-bit aligned address. 1 A value of 1 would indicate that a single bit error occurred in upper 64bits of a 128-bit aligned address and the corrected value was 0. Note : This flag is cleared by writing a 1 to CERRCLR.CPURDERR flag CERRSET.CPURDERR has no effect on this flag. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	FAIL_1_L	R	0h	Fail on 1. 0 Fail on 1 single bit error did not occur in lower 64bits of a 128-bit aligned address. 1 A value of 1 would indicate that a single bit error occurred in lower 64bits of a 128-bit aligned address and the corrected value was 1. Note : This flag is cleared by writing a 1 to CERRCLR.CPURDERR flag CERRSET.CPURDERR has no effect on this flag. Reset type: SYSRSn
0	FAIL_0_L	R	0h	Fail on 0. 0 Fail on 0 single bit error did not occur in lower 64bits of 128-bit data. 1 Would indicate that a single bit error occurred in lower 64bits of a 128-bit aligned address and the corrected value was 0. Note : This flag is cleared by writing a 1 to CERRCLR.CPURDERR flag CERRSET.CPURDERR has no effect on this flag. Reset type: SYSRSn

3.15.14.8 CERRFLG Register (Offset = 20h) [Reset = 00000000h]

CERRFLG is shown in Figure 3-152 and described in Table 3-180.

Return to the Summary Table.

Correctable Error Flag Register

Figure 3-152. CERRFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	DMA_READ_ERR	CPU_READ_ERR
R-0-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 3-180. CERRFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	DMA_READ_ERR	R	0h	DMA correctable Read Error Flag 0: No Error. 1: correctable error occurred during DMA read. Reset type: SYSRSn
0	CPU_READ_ERR	R	0h	CPU correctable Read Error Flag 0: No Error. 1: correctable error occurred during CPU read. Reset type: SYSRSn

3.15.14.9 CERRSET Register (Offset = 22h) [Reset = 00000000h]

CERRSET is shown in Figure 3-153 and described in Table 3-181.

Return to the Summary Table.

Correctable Error Flag Set Register

Figure 3-153. CERRSET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	DMA_READ_ERR_SET	CPU_READ_ERR_SET
R-0-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 3-181. CERRSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	RESERVED	R-0/W1S	0h	Reserved
3	RESERVED	R-0/W1S	0h	Reserved
2	RESERVED	R-0/W1S	0h	Reserved
1	DMA_READ_ERR_SET	R-0/W1S	0h	DMA correctable Read Error Flag Set 0: No Error. 1: DMA Read Error Flag in CERRFLG register will be set and interrupt will be generated if enabled Reset type: SYSRSn
0	CPU_READ_ERR_SET	R-0/W1S	0h	CPU correctable Read Error Flag Set 0: No Error. 1:CPU Read Error Flag in CERRFLG register will be set and interrupt will be generated if enabled Reset type: SYSRSn

3.15.14.10 CERRCLR Register (Offset = 24h) [Reset = 00000000h]

CERRCLR is shown in Figure 3-154 and described in Table 3-182.

Return to the Summary Table.

Correctable Error Flag Clear Register

Figure 3-154. CERRCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	DMA_READ_ERR_CLR	CPU_READ_ERR_CLR
R-0-0h		R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 3-182. CERRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	RESERVED	R-0/W1C	0h	Reserved
4	RESERVED	R-0/W1C	0h	Reserved
3	RESERVED	R-0/W1C	0h	Reserved
2	RESERVED	R-0/W1C	0h	Reserved
1	DMA_READ_ERR_CLR	R-0/W1C	0h	DMA correctable Read Error Flag clear 0: No Error. 1: DMA Read error Flag in CERRFLG register will be cleared Reset type: SYSRSn
0	CPU_READ_ERR_CLR	R-0/W1C	0h	CPU correctable Read Error Flag clear 0: No Error. 1: CPU Read error Flag in CERRFLG register will be cleared Reset type: SYSRSn

3.15.14.11 CCPUREADDR Register (Offset = 26h) [Reset = 00000000h]

CCPUREADDR is shown in Figure 3-155 and described in Table 3-183.

Return to the Summary Table.

Correctable CPU Error Address

Figure 3-155. CCPUREADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCPUREADDR																															
R-0h																															

Table 3-183. CCPUREADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CCPUREADDR	R	0h	This register captures the address location for which CPU read/fetch access resulted in correctable ECC/Parity error. Reset type: SYSRSn

3.15.14.12 CDMAREADDR Register (Offset = 28h) [Reset = 00000000h]

CDMAREADDR is shown in Figure 3-156 and described in Table 3-184.

Return to the Summary Table.

Correctable DMA Error Address

Figure 3-156. CDMAREADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDMAREADDR																															
R-0h																															

Table 3-184. CDMAREADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CDMAREADDR	R	0h	This register captures the address location for which DMA read/fetch access resulted in correctable ECC/Parity error. Reset type: SYSRSn

3.15.14.13 CERRCNT Register (Offset = 30h) [Reset = 00000000h]

CERRCNT is shown in Figure 3-157 and described in Table 3-185.

Return to the Summary Table.

Correctable Error Count Register

Figure 3-157. CERRCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CERRCNT																															
R-0h																															

Table 3-185. CERRCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CERRCNT	R	0h	This register holds the count of how many times correctable error occurred. Reset type: SYSRSn

3.15.14.14 CERRTHRES Register (Offset = 32h) [Reset = 00000000h]

CERRTHRES is shown in Figure 3-158 and described in Table 3-186.

Return to the Summary Table.

Correctable Error Threshold Value Register

Figure 3-158. CERRTHRES Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CERRTHRES															
R-0-0h																R/W-0h															

Table 3-186. CERRTHRES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CERRTHRES	R/W	0h	When value in CERRCNT register is greater than value configured in this register, corretable interrupt gets generated, if enabled. Reset type: SYSRSn

3.15.14.15 CEINTFLG Register (Offset = 34h) [Reset = 00000000h]

CEINTFLG is shown in Figure 3-159 and described in Table 3-187.

Return to the Summary Table.

Correctable Error Interrupt Flag Status Register

Figure 3-159. CEINTFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							CEINTFLAG
R-0-0h							R-0h

Table 3-187. CEINTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	CEINTFLAG	R	0h	Total corrected error count exceeded threshold Flag 0: Total correctable errors < Threshold value configured in CERR_THR register. 1: Total correctable errors >= Threshold value configured in CERR_THR register. Reset type: SYSRSn

3.15.14.16 CEINTCLR Register (Offset = 36h) [Reset = 00000000h]

CEINTCLR is shown in Figure 3-160 and described in Table 3-188.

Return to the Summary Table.

Correctable Error Interrupt Flag Clear Register

Figure 3-160. CEINTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							CEINTCLR
R-0-0h							R-0/W1S-0h

Table 3-188. CEINTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	CEINTCLR	R-0/W1S	0h	0: No action. 1: Total corrected error count exceeded flag in CEINTFLG register will be cleared. Reset type: SYSRSn

3.15.14.17 CEINTSET Register (Offset = 38h) [Reset = 00000000h]

CEINTSET is shown in Figure 3-161 and described in Table 3-189.

Return to the Summary Table.

Correctable Error Interrupt Flag Set Register

Figure 3-161. CEINTSET Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							CEINTSET
R-0-0h							R-0/W1S-0h

Table 3-189. CEINTSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	CEINTSET	R-0/W1S	0h	0: No action. 1: Total corrected error count exceeded flag in CEINTFLG register will be set and interrupt will be generated if enabled. Reset type: SYSRSn

3.15.14.18 CEINTEN Register (Offset = 3Ah) [Reset = 00000000h]

CEINTEN is shown in Figure 3-162 and described in Table 3-190.

Return to the Summary Table.

Correctable Error Interrupt Enable Register

Figure 3-162. CEINTEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							CEINTEN
R-0-0h							R/W-0h

Table 3-190. CEINTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	CEINTEN	R/W	0h	0: Correctable Error Interrupt is disabled. 1: Correctable Error Interrupt is enabled. Reset type: SYSRSn

3.15.15 ROM_WAIT_STATE_REGS Registers

Table 3-191 lists the memory-mapped registers for the ROM_WAIT_STATE_REGS registers. All register offset addresses not listed in Table 3-191 should be considered as reserved locations and the register contents should not be modified.

Table 3-191. ROM_WAIT_STATE_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	ROMWAITSTATE	ROM Wait State Configuration Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 3-192 shows the codes that are used for access types in this section.

Table 3-192. ROM_WAIT_STATE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.15.1 ROMWAITSTATE Register (Offset = 0h) [Reset = 00000000h]

ROMWAITSTATE is shown in Figure 3-163 and described in Table 3-193.

Return to the Summary Table.

ROM Wait State Configuration Register

Figure 3-163. ROMWAITSTATE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							WSDISABLE
R-0h							R/W-0h

Table 3-193. ROMWAITSTATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-1	RESERVED	R	0h	Reserved
0	WSDISABLE	R/W	0h	0: ROM Wait State is enabled. CPU accesses to ROM are are 1-wait. 1: ROM Wait State is disabled. CPU accesses to ROM are 0-wait. Reset type: SYSRSn

3.15.16 TEST_ERROR_REGS Registers

Table 3-194 lists the memory-mapped registers for the TEST_ERROR_REGS registers. All register offset addresses not listed in Table 3-194 should be considered as reserved locations and the register contents should not be modified.

Table 3-194. TEST_ERROR_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	CPU_RAM_TEST_ERROR_STS	Ram Test: Error Status Register	
2h	CPU_RAM_TEST_ERROR_STS_CLR	Ram Test: Error Status Clear Register	
4h	CPU_RAM_TEST_ERROR_ADDR	Ram Test: Error address register	

Complex bit access types are encoded to fit into small table cells. Table 3-195 shows the codes that are used for access types in this section.

Table 3-195. TEST_ERROR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W1C	W 1C	Write 1 to clear
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

3.15.16.1 CPU_RAM_TEST_ERROR_STS Register (Offset = 0h) [Reset = 00000000h]

CPU_RAM_TEST_ERROR_STS is shown in Figure 3-164 and described in Table 3-196.

Return to the Summary Table.

Ram Test: Error Status Register

Figure 3-164. CPU_RAM_TEST_ERROR_STS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						UNC_ERROR	RESERVED
R-0-0h						R-0h	R-0-0h

Table 3-196. CPU_RAM_TEST_ERROR_STS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	UNC_ERROR	R	0h	0: Indicates that there were no 'un-correctable errors' generated in the RAM/ROM test mode. 1: Indicates that 'un-correctable errors' wer generated in the RAM/ROM test mode. Reset type: SYSRSn
0	RESERVED	R-0	0h	Reserved

3.15.16.2 CPU_RAM_TEST_ERROR_STS_CLR Register (Offset = 2h) [Reset = 00000000h]

CPU_RAM_TEST_ERROR_STS_CLR is shown in Figure 3-165 and described in Table 3-197.

Return to the Summary Table.

Ram Test: Error Status Clear Register

Figure 3-165. CPU_RAM_TEST_ERROR_STS_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						UNC_ERROR	RESERVED
R-0-0h						R-0/W1C-0h	R-0-0h

Table 3-197. CPU_RAM_TEST_ERROR_STS_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	UNC_ERROR	R-0/W1C	0h	0: No effect. 1: Clears the corresponding bit in CPU_RAM_TEST_ERROR_STS register. Reset type: SYSRSn
0	RESERVED	R-0	0h	Reserved

3.15.16.3 CPU_RAM_TEST_ERROR_ADDR Register (Offset = 4h) [Reset = 00000000h]

CPU_RAM_TEST_ERROR_ADDR is shown in Figure 3-166 and described in Table 3-198.

Return to the Summary Table.

Ram Test: Error address register

Figure 3-166. CPU_RAM_TEST_ERROR_ADDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
R-0h																															

Table 3-198. CPU_RAM_TEST_ERROR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	Address of the location where error was detected in RAM/ROM test modes. Reset type: SYSRSn

3.15.17 UID_REGS Registers

Table 3-199 lists the memory-mapped registers for the UID_REGS registers. All register offset addresses not listed in Table 3-199 should be considered as reserved locations and the register contents should not be modified.

Table 3-199. UID_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	UID_PSRAND0	UID Psuedo-random 160 bit number	
2h	UID_PSRAND1	UID Psuedo-random 160 bit number	
4h	UID_PSRAND2	UID Psuedo-random 160 bit number	
6h	UID_PSRAND3	UID Psuedo-random 160 bit number	
8h	UID_PSRAND4	UID Psuedo-random 160 bit number	
Ah	UID_UNIQUE0	UID Unique 64 bit number	
Ch	UID_UNIQUE1	UID Unique 64 bit number	
Eh	UID_CHECKSUM	UID Checksum	

Complex bit access types are encoded to fit into small table cells. Table 3-200 shows the codes that are used for access types in this section.

Table 3-200. UID_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-n		Value after reset or the default value

3.15.17.1 UID_PSRAND0 Register (Offset = 0h) [Reset = X0000000h]

UID_PSRAND0 is shown in Figure 3-167 and described in Table 3-201.

Return to the Summary Table.

UID Psuedo-random 160 bit number

Figure 3-167. UID_PSRAND0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RandomID																															
R-Xh																															

Table 3-201. UID_PSRAND0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RandomID	R	Xh	Psuedorandom portion of the UID Reset type: N/A

3.15.17.2 UID_PSRAND1 Register (Offset = 2h) [Reset = X0000000h]

UID_PSRAND1 is shown in Figure 3-168 and described in Table 3-202.

Return to the Summary Table.

UID Psuedo-random 160 bit number

Figure 3-168. UID_PSRAND1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RandomID																															
R-Xh																															

Table 3-202. UID_PSRAND1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RandomID	R	Xh	Psuedorandom portion of the UID Reset type: N/A

3.15.17.3 UID_PSRAND2 Register (Offset = 4h) [Reset = X0000000h]

UID_PSRAND2 is shown in Figure 3-169 and described in Table 3-203.

Return to the Summary Table.

UID Psuedo-random 160 bit number

Figure 3-169. UID_PSRAND2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RandomID																															
R-Xh																															

Table 3-203. UID_PSRAND2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RandomID	R	Xh	Psuedorandom portion of the UID Reset type: N/A

3.15.17.4 UID_PSRAND3 Register (Offset = 6h) [Reset = X0000000h]

UID_PSRAND3 is shown in Figure 3-170 and described in Table 3-204.

Return to the Summary Table.

UID Psuedo-random 160 bit number

Figure 3-170. UID_PSRAND3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RandomID																															
R-Xh																															

Table 3-204. UID_PSRAND3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RandomID	R	Xh	Psuedorandom portion of the UID Reset type: N/A

3.15.17.5 UID_PSRAND4 Register (Offset = 8h) [Reset = X0000000h]

UID_PSRAND4 is shown in Figure 3-171 and described in Table 3-205.

Return to the Summary Table.

UID Psuedo-random 160 bit number

Figure 3-171. UID_PSRAND4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RandomID																															
R-Xh																															

Table 3-205. UID_PSRAND4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RandomID	R	Xh	Psuedorandom portion of the UID Reset type: N/A

3.15.17.6 UID_UNIQUE0 Register (Offset = Ah) [Reset = X0000000h]

UID_UNIQUE0 is shown in Figure 3-172 and described in Table 3-206.

Return to the Summary Table.

UID Unique 64 bit number

Figure 3-172. UID_UNIQUE0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UniqueID																															
R-Xh																															

Table 3-206. UID_UNIQUE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UniqueID	R	Xh	Unique portion of the UID. This identifier will be unique across all devices with the same PARTIDH. Reset type: N/A

3.15.17.7 UID_UNIQUE1 Register (Offset = Ch) [Reset = X0000000h]

UID_UNIQUE1 is shown in Figure 3-173 and described in Table 3-207.

Return to the Summary Table.

UID Unique 64 bit number

Figure 3-173. UID_UNIQUE1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UniqueID																															
R-Xh																															

Table 3-207. UID_UNIQUE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UniqueID	R	Xh	Unique portion of the UID. This identifier will be unique across all devices with the same PARTIDH. Reset type: N/A

3.15.17.8 UID_CHECKSUM Register (Offset = Eh) [Reset = X0000000h]

UID_CHECKSUM is shown in Figure 3-174 and described in Table 3-208.

Return to the Summary Table.

Fletcher checksum of UID_PSRAND and UID_UNIQUE registers

Figure 3-174. UID_CHECKSUM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Checksum																															
R-Xh																															

Table 3-208. UID_CHECKSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Checksum	R	Xh	Fletcher checksum of UID_PSRANDx and UID_UINIQUE Reset type: N/A

Chapter 4

ROM Code and Peripheral Booting

This chapter explains the boot procedure, the available boot modes, and the various details of the ROM code including memory maps, initializations, reset handling, and status information.

4.1 Introduction

The purpose of this chapter is to explain the boot read-only memory (ROM) code functionality for the CPU core, including the boot procedure. This chapter also discusses the functions and features of the boot ROM code, and provides details about the ROM memory-map contents. On every reset, the device executes a boot sequence in the ROM depending on the reset type and boot configuration. This sequence initializes the device to run the application code. For the CPU, the boot ROM also contains peripheral bootloaders that can be used to load an application into RAM. These bootloaders can be disabled for safety or security purposes.

See Table 4-1 for details on available boot features for the C28x CPU. Additionally, Table 4-2 shows the sizes of the various ROMs on the device.

For details on the security APIs provided, refer to Section 4.7.9.

Various tables are provided in ROM for use in software library, refer to Section 4.7.6 for more details.

Table 4-1. Boot System Overview

Boot Feature	CPU
Initial boot process	Device reset
Boot mode selection	GPIOs
Boot modes supported	Flash boot Secure Flash boot RAM boot
Peripheral boot loaders supported	Parallel IO SCI / Wait I2C SPI

Table 4-2. ROM Memory

ROM	CPU Size
Secure and Unsecure boot ROM	64KB

4.1.1 ROM Related Collateral

Foundational Materials

- Bootloading 101 (Video)

Getting Started Materials

- *Secure BOOT On C2000 Device Application Report*

Expert Materials

- *C2000 Software Controlled Firmware Update Process Application Report*

4.2 Device Boot Sequence

Table 4-3 describes the general boot ROM procedure each time the CPU core is reset.

During boot, boot ROM code updates a boot status location in RAM that details the actions taken during this process. Refer to Section 4.7.11 for more details.

Table 4-3. Device Boot ROM Sequence

Step	CPU Action
1	After reset, perform watchdog initialization
2	Clock configuration and Flash power-up
3	Peripheral trimming and device configuration registers are loaded from OTP.
4	On power-on reset (POR), all RAMs are initialized.
5	Non-maskable interrupt (NMI) handling is enabled and DCSM initialization is performed.
6	Device calibration is performed; trimming the specified peripherals with set OTP values.
7	Determine if polling the GPIO pins are needed for determining the boot mode and, if so, read the boot mode GPIO pins to determine the boot mode to run.
8	Based on the boot mode and options, the appropriate boot sequence is executed. Refer to Section 4.5.1 for a flow chart of the boot sequences.

4.3 Device Boot Modes

This section explains the default boot modes, as well as all the available custom boot modes supported on this device. The boot ROM uses the boot mode select, general purpose input/output (GPIO) pins to determine the boot mode configuration.

4.3.1 Default Boot Modes

Table 4-4 shows the boot mode options available for selection by the default boot mode select pins. Users have the option to program the device to customize the boot modes selectable in the boot-up table as well as the boot mode select pin GPIOs used.

Table 4-4. Device Default Boot Modes

Boot Mode	GPIO24 (Default boot mode select pin 1)	GPIO32 (Default boot mode select pin 0)
Parallel IO	0	0
SCI / Wait Boot(1)	0	1
Flash	1	1

(1) SCI boot mode is used as a wait boot mode as long as SCI continues to wait for an 'A' or 'a' during the SCI autobaud lock process.

Refer to Section 4.7.7.1 for functional details of the boot modes.

Refer to Section 4.7.8 for GPIOs used for selecting the boot modes.

Refer to Section 4.4 for details of boot configurations.

Note

All the peripheral boot modes that are supported use the first instance of the peripheral module (SCIA, SPIA, I2CA, and so forth). Whenever these boot modes are referred to in this chapter, such as SCI boot, the mode is actually referring to the first module instance, which means the SCI boot on the SCIA port. The same applies to the other peripheral boot modes.

4.3.2 Custom Boot Modes

Once the user programs a custom boot table in user OTP, an entry in the custom table is used for booting. Users can customize the boot mode select pins in the end system design by programming the BOOTPIN_CONFIG location in user OTP. This allows customers to use 0, 1, 2, or 3 boot mode select pins as needed. You can also customize the boot definition table and indicate which location to boot from by programming the boot mode definition table in the BOOTDEF location of user OTP. Table 4-5 shows the options for various boot modes.

Table 4-5. Custom Boot Modes

Boot Mode Number	Boot Modes
0x0	Parallel
0x1	SCI / Wait
0x3	Flash
0x4	Wait
0x5	RAM
0x6	SPI
0x7	I2C
0xA	Secure Flash

4.4 Device Boot Configurations

This section details what boot configurations are available and how to configure them. This device supports from zero boot mode select pins up to three boot mode select pins and from one configured boot mode up to eight configured boot modes.

To change and configure the device from the default settings to custom settings for your application, use the following process:

1. Determine all the various ways you want application to be able to boot. (For example: Primary boot option of Flash boot for your main application, secondary boot option of SPI for firmware updates, tertiary boot option of SCI boot for debugging, etc)
2. Based on the number of boot modes needed, determine how many boot mode select pins (BMSPs) are required to select between your selected boot modes. (For example: Two BMSPs are required to select between three boot mode options)
3. Assign the required BMSPs to a physical GPIO pin. (For example, BMSP0 to GPIO10, BMSP1 to GPIO51, and BMSP2 left as default which is disabled). Refer to Section 4.4.1 for all the details on performing these configurations.
4. Assign the determined boot mode definitions to indexes in your custom boot table that correlate to the decoded value of the BMSPs. For example, BOOTDEF0=Boot to Flash, BOOTDEF1=SPI Boot, BOOTDEF2=SCI Boot; all other BOOTDEFx are left as default/nothing). Refer to Section 4.4.2 for all the details on setting up and configuring the custom boot mode table.

Additionally, Section 4.4.3 provides some example use cases on how to configure the BMSPs and custom boot tables.

4.4.1 Configuring Boot Mode Pins

This section explains how the boot mode select pins are customized by the user, by programming the BOOTPIN-CONFIG location (refer to Table 4-6), in the user-configurable dual-zone security module (DCSM) OTP. The location in the DCSM OTP is Z1-OTP-BOOTPIN-CONFIG or Z2-OTP-BOOTPIN-CONFIG. When debugging, EMU-BOOTPIN-CONFIG is the emulation equivalent of Z1-OTP-BOOTPIN-CONFIG/Z2-OTP-BOOTPIN-CONFIG, and can be programmed to experiment with different boot modes without writing to OTP. The device can be programmed to use **zero**, **one**, **two**, or **three** boot mode select pins as needed.

Note

When using Z2-OTP-BOOTPIN-CONFIG, the configurations programmed in this location take priority over the configurations in Z1-OTP-BOOTPIN-CONFIG. It is recommended to use Z1-OTP-BOOTPIN-CONFIG first and then, if OTP configurations need to be altered, switch to using Z2-OTP-BOOTPIN-CONFIG.

Table 4-6. BOOTPIN-CONFIG Bit Fields

Bit	Name	Description
31:24	Key	Write 0x5A to these 8-bits to tell the boot ROM code that the bits in this register are valid.
23:16	Boot Mode Select Pin 2 (BMSP2)	Refer to BMSP0 description.
15:8	Boot Mode Select Pin 1 (BMSP1)	Refer to BMSP0 description.
7:0	Boot Mode Select Pin 0 (BMSP0)	Set to the GPIO pin to be used during boot (up to 255). 0x0 = GPIO0, 0x01 = GPIO1, and so on. Writing 0xFF disables this BMSP and this pin is no longer used to select the boot mode.

Note

GPIO 12, 13, 20, 21, 28, 224, 226-228, 230, 242-243 are analog pins, but digital inputs are possible on these pins provided the software writes to the GPIOHAMSEL register bits.

The following GPIOs cannot be used as a BMSP. If selected for a particular BMSP, the boot ROM automatically selects the factory default GPIOs for BMSP0 and BMSP1. Factory default for BMSP2 is 0xFF, which disables the BMSP.

- GPIO 8, GPIO 14, and GPIO 15 (Not available on any package)
- GPIO 17 to GPIO 22 (Not available on any package)
- GPIO 25 to GPIO 27 (Not available on any package)
- GPIO 31, GPIO 34 to GPIO 38 (Not available on any package)
- GPIO 42, GPIO 44, and GPIO 46 to GPIO 49 (Not available on any package)
- GPIO 225 and GPIO 229 (Not available on any package)
- GPIO 231 to GPIO 241 (Not available on any package)
- GPIO 244 and GPIO 245 (Not available on any package)

Table 4-7. Standalone Boot Mode Select Pin Decoding

BOOTPIN_CONFIG Key	BMSP0	BMSP1	BMSP2	Realized Boot Mode
!= 0x5A	Don't Care	Don't Care	Don't Care	Boot as defined by the factory default BMSPs.
= 0x5A	0xFF	0xFF	0xFF	Boot as defined in the boot table for boot mode 0 (All BMSPs disabled).
	Valid GPIO	0xFF	0xFF	Boot as defined by the value of BMSP0 (BMSP1 and BMSP2 disabled).
	0xFF	Valid GPIO	0xFF	Boot as defined by the value of BMSP1 (BMSP0 and BMSP2 disabled).
	0xFF	0xFF	Valid GPIO	Boot as defined by the value of BMSP2 (BMSP0 and BMSP1 disabled)
	Valid GPIO	Valid GPIO	0xFF	Boot as defined by the values of BMSP0 and BMSP1 (BMSP2 disabled).
	Valid GPIO	0xFF	Valid GPIO	Boot as defined by the values of BMSP0 and BMSP2 (BMSP1 disabled).
	0xFF	Valid GPIO	Valid GPIO	Boot as defined by the values of BMSP1 and BMSP2 (BMSP0 disabled).
	Valid GPIO	Valid GPIO	Valid GPIO	Boot as defined by the values of BMSP0, BMSP1, and BMSP2.
	Invalid GPIO	Valid GPIO	Valid GPIO	BMSP0 is reset to the factory default BMSP0 GPIO. Boot as defined by the values of BMSP0, BMSP1, and BMSP2.
	Valid GPIO	Invalid GPIO	Valid GPIO	BMSP1 is reset to the factory default BMSP1 GPIO. Boot as defined by the values of BMSP0, BMSP1, and BMSP2.
	Valid GPIO	Valid GPIO	Invalid GPIO	BMSP2 is reset to the factory default state, which is disabled. Boot as defined by the values of BMSP0 and BMSP1.

Note

When decoding the boot mode, BMSP0 is the least-significant bit and BMSP2 is the most-significant bit of the boot table index value. It is recommended when disabling BMSPs to start with disabling BMSP2. For example, in an instance when only using BMSP2 (BMSP1 and BMSP0 are disabled), then only the boot table indexes of 0 and 4 are selectable. In the instance when using only BMSP0, then the selectable boot table indexes are 0 and 1.

4.4.2 Configuring Boot Mode Table Options

This section explains how to configure the boot definition table, BOOTDEF, for the device and the associated boot options. The 64-bit location is located in user-configurable DCSM OTP in the Z1-OTP-BOOTDEF-LOW and Z1-OTP-BOOTDEF-HIGH locations. When debugging, EMU-BOOTDEF-LOW and EMU-BOOTDEF-HIGH are the emulation equivalents of Z1-OTP-BOOTDEF-LOW and Z1-OTP-BOOTDEF-HIGH, and can be programmed to experiment with different boot mode options without writing to OTP. The range of customization to the boot definition table depends on how many boot mode select pins (BMSP) are being used. For example, 0 BMSPs equals to 1 table entry, 1 BMSP equals to 2 table entries, 2 BMSPs equals to 4 table entries, and 3 BMSPs equals to 8 table entries. Refer to Section 4.4.3 for examples on how to setup the BOOTPIN_CONFIG and BOOTDEF values.

Note

The locations Z2-OTP-BOOTDEF-LOW and Z2-OTP-BOOTDEF-HIGH are used instead of Z1-OTP-BOOTDEF-LOW and Z1-OTP-BOOTDEF-HIGH locations when Z2-OTP-BOOTPIN-CONFIG is configured. Refer to Section 4.4.1 for more details on BOOTPIN_CONFIG usage.

Table 4-8. BOOTDEF Bit Fields

BOOTDEF Name	Byte Position	Name	Description
BOOT_DEF0	7:0	[3:0] BOOT_DEF0 Mode	Set the boot mode number from Section 4.3.2. Any unsupported boot mode causes the device to either go to Wait boot (debugger connected) or boot to Flash (standalone).
		[7:4] BOOT_DEF0 Options	Set alternate/additional boot options. This can include changing the GPIOs for a particular boot peripheral or specifying a different Flash entry point. Refer to Section 4.7.8 for valid BOOTDEF values to set in the table.
BOOT_DEF1	15:8	BOOT_DEF1 Mode/ Options	Refer to BOOT_DEF0 description.
BOOT_DEF2	23:16	BOOT_DEF2 Mode/ Options	
BOOT_DEF3	31:24	BOOT_DEF3 Mode/ Options	
BOOT_DEF4	39:32	BOOT_DEF4 Mode/ Options	
BOOT_DEF5	47:40	BOOT_DEF5 Mode/ Options	
BOOT_DEF6	55:48	BOOT_DEF6 Mode/ Options	
BOOT_DEF7	63:56	BOOT_DEF7 Mode/ Options	

4.4.3 Boot Mode Example Use Cases

This section demonstrates some use cases for configuring the boot mode select pins and boot modes.

4.4.3.1 Zero Boot Mode Select Pins

This use case demonstrates a scenario for an application that does not use any boot mode select pins and always has the device boot to Flash.

1. Program the BOOTPIN_CONFIG location in OTP as follows:
   - Set BOOTPIN_CONFIG.BMSP0 to 0xFF
   - Set BOOTPIN_CONFIG.BMSP1 to 0xFF
   - Set BOOTPIN_CONFIG.BMSP2 to 0xFF
   - Set BOOTPIN_CONFIG.KEY to 0x5A for boot ROM to treat these register bits as valid and use the custom boot table.
2. Program the BOOTDEF location options for the device. This essentially sets up a device-specific boot mode table. Refer to Section 4.7.8 for valid BOOTDEF values to set in the table.
   - Set BOOTDEF.BOOTDEF0 to 0x03 for booting to Flash (entry address option 0). This sets Flash boot to boot table index 0.
   - Refer to Section 4.7.2 for the available Flash entry points.

Table 4-9. Zero Boot Pin Boot Table Result

Boot Mode Table Number	Boot Mode
0	Flash Boot (0x03)

4.4.3.2 One Boot Mode Select Pin

This use case demonstrates a scenario for an application using one boot mode select pin to select between booting to Flash and using SCI boot.

1. Program the BOOTPIN_CONFIG location in OTP as follows:
   - Set BOOTPIN_CONFIG.BMSP0 to a user specified GPIO, such as 0x0 for GPIO0
   - Set BOOTPIN_CONFIG.BMSP1 to 0xFF
   - Set BOOTPIN_CONFIG.BMSP2 to 0xFF
   - Set BOOTPIN_CONFIG.KEY to 0x5A for boot ROM to treat these register bits as valid and use the custom boot table.
2. Program the BOOTDEF location options for the device. This essentially sets up a device-specific boot mode table. Refer to Section 4.7.8 for valid BOOTDEF values to set in the table.
   - Set BOOTDEF.BOOTDEF0 to 0x03 for booting to Flash (entry address option 0). This sets Flash boot to boot table index 0.
   - Set BOOTDEF.BOOTDEF1 to 0x01 for SCI booting. This sets SCI boot to boot table index 1.

Table 4-10. One Boot Pin Boot Table Result

Boot Mode Table Number	Boot Mode
0	Flash Boot (0x03)
1	SCI Boot (0x01)

4.4.3.3 Three Boot Mode Select Pins

This use case demonstrates a scenario for an application using three boot mode select pins to select between various boot modes in the custom boot table.

1. Program the BOOTPIN_CONFIG location in OTP as follows:
   - Set BOOTPIN_CONFIG.BMSP0 to a user specified GPIO, such as 0x0 for GPIO0
   - Set BOOTPIN_CONFIG.BMSP1 to a user specified GPIO, such as 0x1 for GPIO1
   - Set BOOTPIN_CONFIG.BMSP2 to a user specified GPIO, such as 0x2 for GPIO2
   - Set BOOTPIN_CONFIG.KEY to 0x5A for boot ROM to treat these register bits as valid and use the custom boot table.
2. Program the BOOTDEF location options for the device. This essentially sets up a device-specific boot mode table. Refer to Section 4.7.8 for valid BOOTDEF values to set in the table.
   - Set BOOTDEF.BOOTDEF1 to 0x03 for booting to Flash (entry address option 0). This sets Flash boot to boot table index 1.
   - Set BOOTDEF.BOOTDEF2 to 0x24 for booting to wait boot (alternate option). This sets wait boot to boot table index 2.
   - Set BOOTDEF.BOOTDEF3 to 0x66 for SPI booting (alternate GPIO option 3). This sets SPI boot to boot table index 3.
   - Set BOOTDEF.BOOTDEF4 to 0x43 for booting to Flash (entry address option 2). This sets Flash boot to boot table index 4.

Table 4-11. Three Boot Pins Boot Table Result

Boot Mode Table Number	Boot Mode
1	Flash Boot (0x03)
2	Wait Boot - Alt (0x24)
3	SPI - Alt3 (0x66)
4	Flash Boot - Alt2 (0x43)
5, 6, 7	Not used in this example

4.5 Device Boot Flow Diagrams

This section details the boot flow diagrams for standalone and emulation boot flows.

4.5.1 Boot Flow

Upon reset, the CPU follows the boot flow shown in Figure 4-1. Depending on whether a JTAG debugger is connected to the device, the CPU either continues booting following the emulation boot flow or the standalone boot flow.

Note

BOR follows same flow as POR.

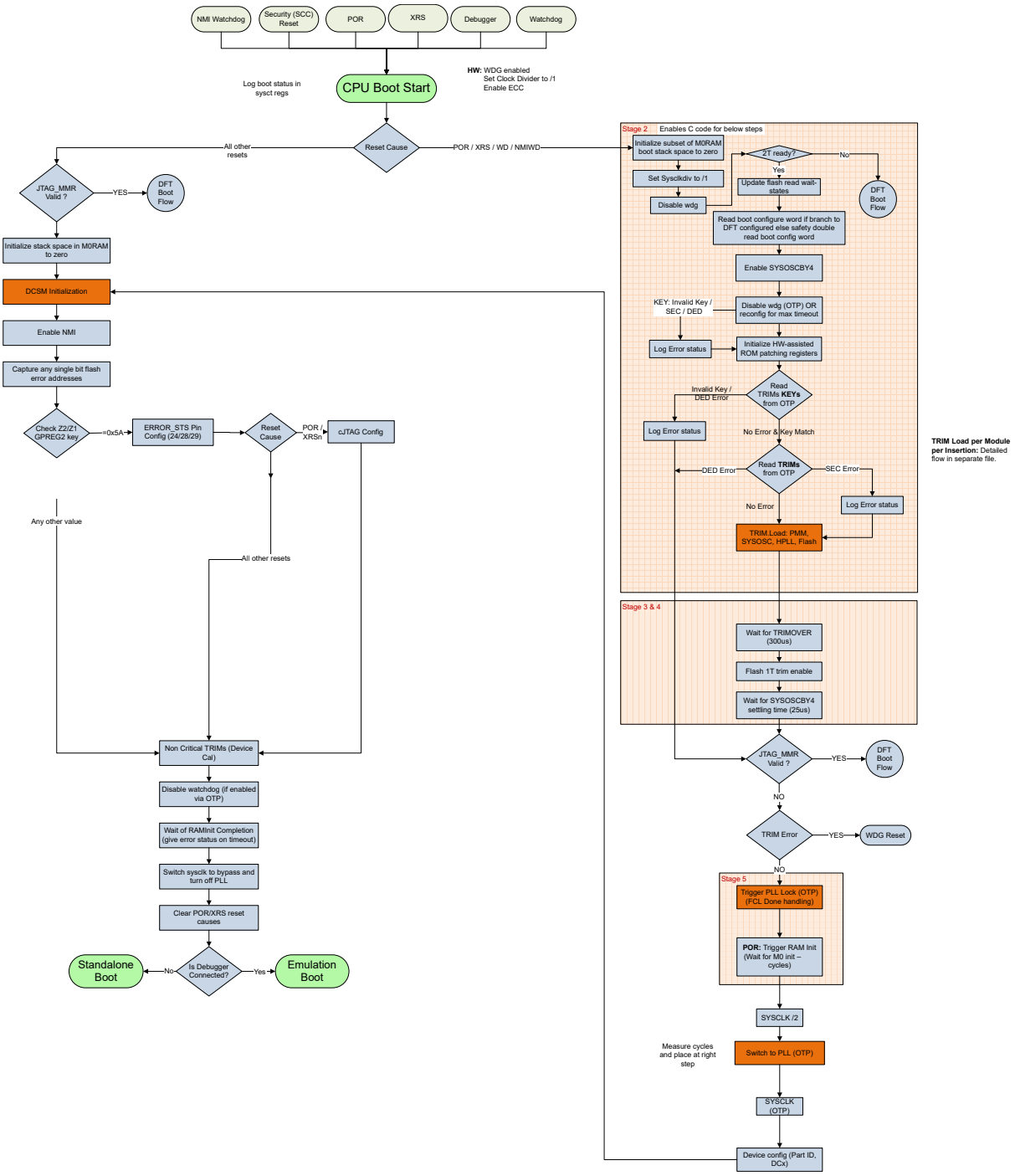


Figure 4-1. Device Boot Flow

4.5.2 Emulation Boot Flow

Figure 4-2 shows the emulation boot flow when JTAG debugger is connected.

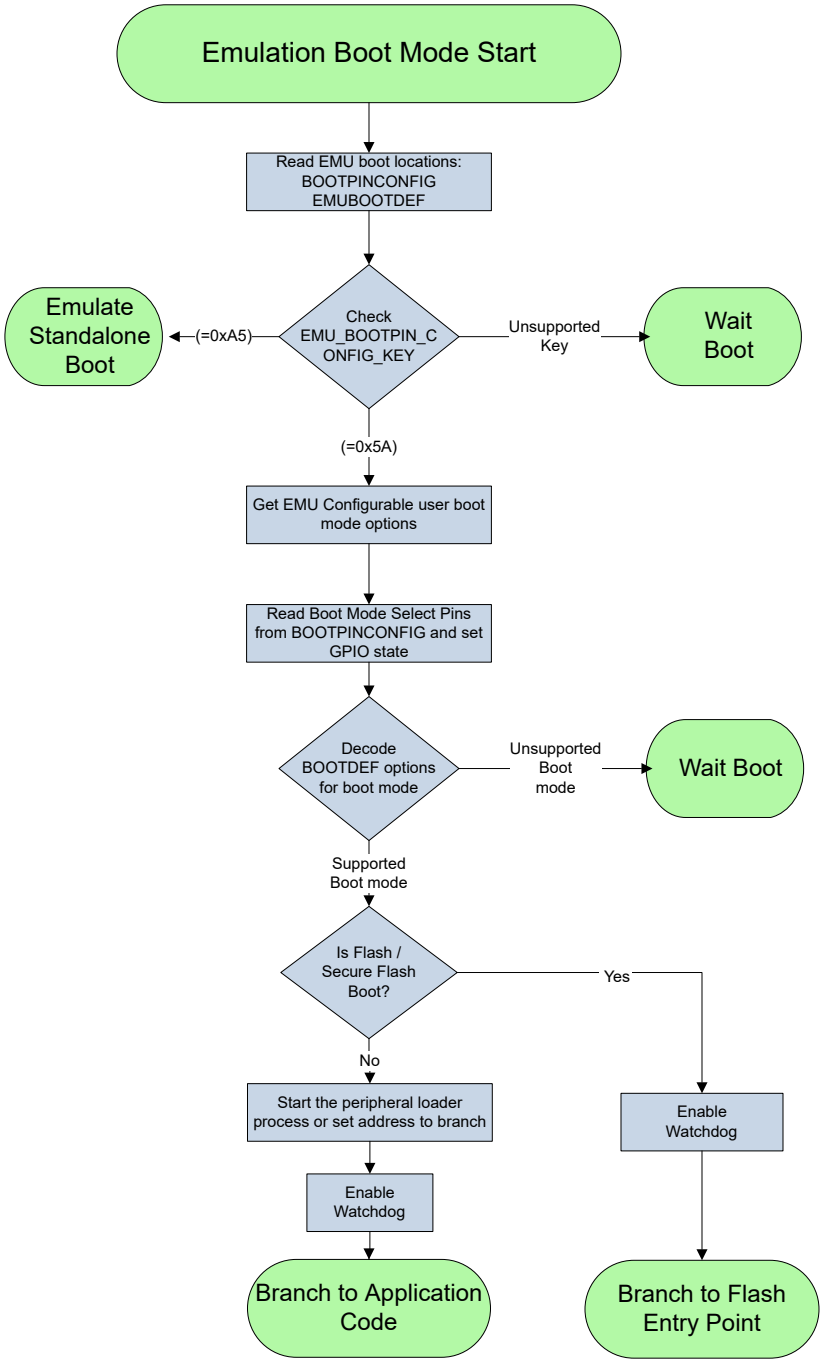


Figure 4-2. Emulation Boot Flow

4.5.3 Standalone Boot Flow

Figure 4-3 shows the standalone boot flow when no JTAG debugger is connected to the device.

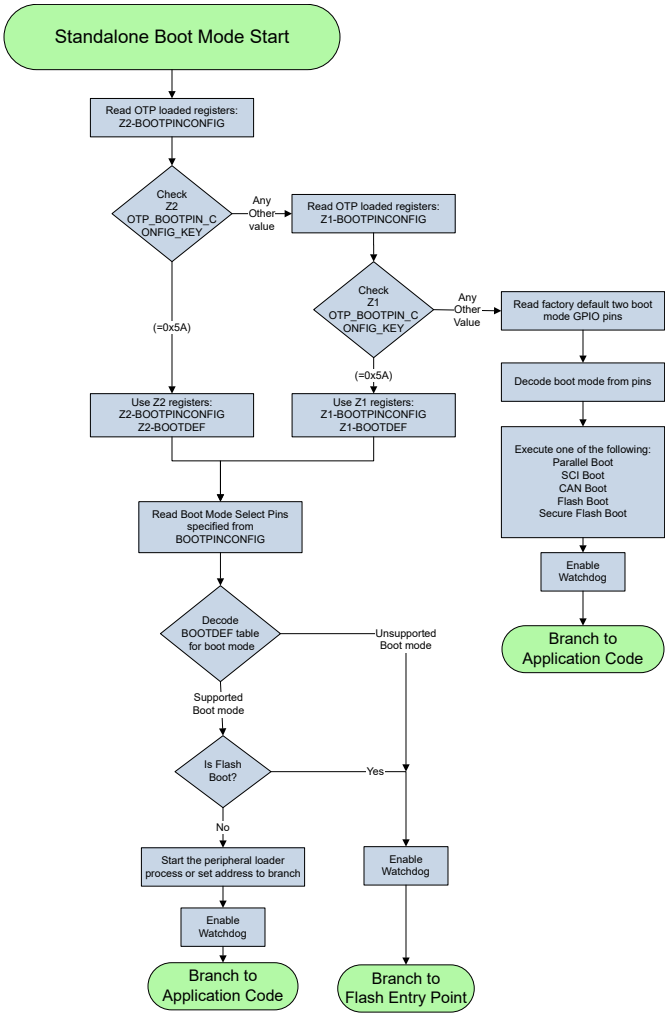


Figure 4-3. CPU Standalone Boot Flow

4.6 Device Reset and Exception Handling

4.6.1 Reset Causes and Handling

Table 4-12 explains the actions each boot ROM performs upon reset for a specific reset cause.

Table 4-12. Boot ROM Reset Causes and Actions

Reset Source	Boot ROM Action
Power on Reset (POR)	1. Configure Clock Divider 2. Flash Power Up 3. Device configuration and trimming 4. RAM Initialization 5. Continue default boot flow
External Reset (XRS) Includes: • Watchdog Reset • NMI Watchdog Reset • SIMRESET XRS	1. Configure Clock Divider 2. Flash Power Up 3. Device configuration and trimming 4. Clear RAM for boot stack 5. Continue default boot flow
Secure Copy Code (SCC) Reset	1. Clear RAM for boot stack 2. Continue default boot flow
SIMRESET	1. Clear RAM for boot stack 2. Continue default boot flow
Debugger Reset	1. Clear RAM for boot stack 2. Continue default boot flow

4.6.2 Exceptions and Interrupts Handling

Table 4-13 explains the actions that the boot ROM performs, if any exceptions occur during boot. The exception handling philosophy in most cases is to log the error and continue booting to reach the application.

Table 4-13. Boot ROM Exceptions and Actions

Exception Event Source	Boot ROM Action	Event Logged
Clock Fail	Clear the NMI flag and continue to boot	Yes
RAM Uncorrectable Error ROM Parity Error	Perform RAM initialization and reset the device	Yes(1)
Flash Uncorrectable Error	Reset the device	Yes
Software NMI error (Software self test error)	Reset the device	No
ITRAP Exception	Record memory address of where the illegal instruction was executed and let device reset	Yes
Unsupported PIE Interrupts	Ignore and continue to boot	No

(1) A RAM uncorrectable error or ROM parity error clears the boot status information stored in RAM because a RAM initialization is performed to attempt to correct the error. Since the boot status information is erased, this exception can be identified in that a NMIWD reset occurred and all the RAMs are erased.

4.7 Boot ROM Description

This section explains the details regarding the device boot ROMs.

4.7.1 Boot ROM Configuration Registers

The boot ROM code involves several memory addresses and registers used during execution. There are two sets of configurations; one for emulation and one for standalone boot flow. The emulation locations located in RAM emulate the OTP configurations and can be written to as many times as needed. The user configurable DCSM OTP locations used in the standalone boot flow program the device OTP and hence can only be written once. Table 4-14 details these locations. For bit field configuration details for BOOTPIN-CONFIG and BOOTDEF, see Section 4.4.1 and Section 4.4.2.

Additionally, the boot ROM supports boot configurations from DCSM zone 1 and zone 2 registers. Zone 2 configurations supercede zone 1 configurations, so it is recommended to use zone 1 configurations and use zone 2 as a secondary option.

Table 4-14. Boot ROM Registers

Boot Flow	Register Name	Boot ROM Name	Register Address	User OTP Address
Emulation	-	EMU-BOOTPIN-CONFIG	0x0000 0D00	-
	-	EMU-GPREG2	0x0000 0D02	-
	-	EMU-BOOTDEF-LOW	0x0000 0D04	-
	-	EMU-BOOTDEF-HIGH	0x0000 0D06	-
Standalone (Using Z1)	Z1-GPREG1	Z1-OTP-BOOTPIN-CONFIG	0x0005 F008	0x0007 8008
	Z1-GPREG2	Z1-OTP-BOOT-GPREG2	0x0005 F00A	0x0007 800A
	Z1-GPREG3	Z1-OTP-BOOTDEF-LOW	0x0005 F00C	0x0007 800C
	Z1-GPREG4	Z1-OTP-BOOTDEF-HIGH	0x0005 F00E	0x0007 800E
Standalone (Using Z2)	Z2-GPREG1	Z2-OTP-BOOTPIN-CONFIG	0x0005 F088	0x0007 8208
	Z2-GPREG2	Z2-OTP-BOOT-GPREG2	0x0005 F08A	0x0007 820A
	Z2-GPREG3	Z2-OTP-BOOTDEF-LOW	0x0005 F08C	0x0007 820C
	Z2-GPREG4	Z2-OTP-BOOTDEF-HIGH	0x0005 F08E	0x0007 820E

4.7.1.1 GPREG2 Usage and Configuration

Table 4-15 explains how the bit field values from the user configurable DCSM OTP location, Z1-OTP-BOOT-GPREG2, are decoded by boot ROM.

Table 4-15. DCSM Zone GPREG2 Bit Fields

Bit	Name	Description	Boot ROM Action
31:24	Key	Write 0x5A to indicate to the boot ROM code that the bits in this register are valid.	If user sets to 0x5A, boot ROM uses the values in this register. If set to any other value, boot ROM ignores values in this register.
23:6	Reserved	Reserved	No Action
5:4	ERROR_STS_PIN config	0x0 – GPIO24, MUX Option 13 0x1 – GPIO28, MUX Option 13 0x2 – GPIO29, MUX Option 13 0x3 – ERROR_STS function disabled (default)	This indicates which GPIO pin is supposed to be used as ERROR_PIN and boot ROM configures the mux as such for the said pin. The ERROR_STS pin mux configuration is locked by the boot ROM, but not committed.
3:0	CJTAGNODEID	CJTAGNODEID[3:0]	Boot ROM takes this values and programs the lower 4 bits of the CJTAGNODEID register.

4.7.2 Entry Points

This sections gives details about the entry point addresses for various boot modes. These entry points direct the boot ROM what address to branch to at the end of booting as per the selected boot mode.

Table 4-16 gives details about the entry point addresses for Flash boot mode.

Table 4-17 gives details about the entry point addresses for RAM boot mode.

Table 4-18 gives details about the entry point addresses for Secure Flash boot mode.

Table 4-16. Flash Boot Options

Option	BOOTDEFx Value	Flash Entry Address	Flash Sector	Package Supported
0 (default)	0x03	0x0008 0000	CPU1 Bank 0 Sector 0	All
1	0x23	0x0008 8000	CPU1 Bank 0 Sector 32	All

Table 4-17. RAM Boot Options

Option	BOOTDEFx Value	RAM Entry Point (Address)	Package Supported
0	0x05	0x0000 0000	All

Table 4-18. Secure Flash Boot Options

Option	BOOTDEFx Value	Flash Entry Address	Flash Sector	Package Supported
0 (default)	0x0A	0x0008 0000	CPU1 Bank 0 Sector 0	All
1	0x2A	0x0008 8000	CPU1 Bank 0 Sector 32	All

4.7.3 Wait Points

The wait mode puts the CPU in a loop in the boot ROM code and does not branch to the user application code. The device can enter the wait boot mode either through manually being set or because of some issue during boot up. Using the wait boot mode is recommended when using a debugger to avoid any JTAG issues. There is an ESTOP provided for debugging during Wait boot.

Table 4-19. Wait Boot Options

Option	BOOTDEFx Value	Watchdog Status	Package Supported
0	0x04	Enabled	All
1	0x24	Disabled	All

During boot ROM execution, there are situations where the CPU can enter a wait loop in the code. This state can occur for a variety of reasons. Table 4-20 details the address ranges that the CPU PC register value falls between, if the CPU has entered one of these instances.

Following are the actions for entering wait boot mode:

- Wait boot is set by the user as the boot mode.
- Boot mode is unrecognizable and a debugger is connected to the device.
- The emulation BOOTPIN_CONFIG key is not equal to 0xA5 or 0x5A.
- An error occurs during emulation boot and the boot mode pins are decoded with a value not recognized as a valid boot mode.

Table 4-20. Wait Point Addresses

Address Range	Description
0x003F CCA4 – 0x003F CCAA	In Wait Boot Mode
0x003F D886 – 0x003F D88E	In SCI Boot waiting on autobaud lock
0x003F FE32 – 0x003F FEC4	In NMI Handler
0x003F FDE5 – 0x003F FE32	In PIE Mismatch Handler
0x003F FEC4 – 0x003F FEF6	In ITRAP ISR
0x003F D330 – 0x003F D333 0x003F D24A – 0x003F D260	In Parallel boot waiting for control signal

4.7.4 Secure Flash Boot

Secure Flash boot mode is similar to Flash boot mode in that the boot flow branches to the configured memory address in Flash but only after the Flash memory contents have been authenticated. The Flash authentication uses a Cipher-based Message Authentication Protocol (CMAC) to authenticate 16KB of Flash starting from the configured Flash entry point address. The CMAC calculation requires a user-defined 128-bit key programmed in the CPU User OTP Zone 1 Header OTP CMACKEY bit field. Additionally, you must calculate the golden CMAC tag based on the 16KB Flash memory range and store the tag along with the user code at a hardcoded address in Flash. During secure Flash boot, the calculated CMAC tag is compared to the user golden CMAC tag in Flash to determine the pass/fail status of the CMAC authentication. When authentication passes, boot flow continues and branches to Flash to begin executing the application. When authentication fails, the device is reset.

For the available secure Flash boot entry address options, refer to Section 4.7.2.

For generating the secure Flash golden CMAC tag for CPU, refer to the *TMS320C28x Assembly Language Tools User's Guide* within section "Using Secure Flash Boot on TMS320F2838x Devices" for instructions.

Note

Both the CMAC golden signature and CMAC key are stored in the most significant double format, but each 32-bit section is in little-endian format.

Key: 2B7E 1516 28AE D2A6 ABF7 1588 09CF 4F3C

(MSB is 2B and LSB is 3C)

CMACKEY0 = 0x2B7E 1516

CMACKEY1 = 0x28AE D2A6

CMACKEY2 = 0xABF7 1588

CMACKEY3 = 0x09CF 4F3C

Note

You must make sure that the Flash sector that encompasses the configured Flash entry point and the first 16KB of Flash is assigned to Zone 1 for the core setup for secure Flash boot.

Recommended to use device JTAGLOCK when using secure Flash boot.

Table 4-21. Secure Flash Boot Details

Details	Location Address
CMAC Signature Address	Flash Entry point Address + 0x2
CMAC Key Address (128-bit key)	DCSM Z1 OTP CMACKEY0/1/2/3
Flash Entry Point (Bank 0, Sector 0)	0x0008 0000
Flash Entry Point (Bank 0, Sector 16)	0x0008 8000
Flash Entry Point (Bank 0, End of Sector 32)	0x0009 0000
Flash Entry Point (Bank 0, Sector 48)	0x0009 8000
Address Range for CMAC Calculation	Start: Flash Entry Point Address End: Flash Entry Point Address + 16KB

Table 4-22. Secure Flash Tag and Key Details

Name	Address	Details
CMA Golden Tag (128-bit)	**CPU:** *Flash Entry Point Address* + 0x2	Located in Flash, offset from the entry point address, by 2 words (CPU). When CMAC calculations are performed, the golden tag location in memory is considered all 0xFs. Refer to Example 4-1 for an example regarding linker configuration on CPU. Lower memory contains the tag's MSW and higher memory contains the LSW. **Example (on CPU):** Tag = 0x0011 2233 4455 6677 8899 AABB CCDD EEFF Address 0x0 = 0x0011 2233 Address 0x2 = 0x4455 6677 Address 0x4 = 0x8899 AABB Address 0x6 = 0xCCDD EEFF
CMAC 128-Bit Key	0x0007 8018	Located in CPU Zone 1 User Header OTP (CMACKEY0, CMACKEY1, CMACKEY2, CMACKEY3) CMACKEY0 contains the key's MSW and CMACKEY3 contains the LSW. **Example:** Key = 0x0011 2233 4455 6677 8899 AABB CCDD EEFF CMACKEY0 = 0x0011 2233 CMACKEY1 = 0x4455 6677 CMACKEY2 = 0x8899 AABB CMACKEY3 = 0xCCDD EEFF

Table 4-23. Secure Flash Authentication Failure Actions

CPU	Action on Failed Authentication
C28x CPU	1. Emulation only - Halt debugger (ESTOP) 2. Wait in endless loop (for device reset due to Watchdog reset)

Table 4-24. Secure Flash on all CPUs Recommended Flow

Step	Action
1	Secure Flash boot CPU
2	Any Flash beyond the first 16KB from the entry point that is planned for use can be authenticated by you using a different CMAC golden tag embedded at an address somewhere within the already authenticated 16KB of Flash.

APIs for CMAC calculation and authentication is provided as part of ROM. Details are available in Section 4.7.9.

Example 4-1. Secure Flash CPU1 Linker File Example

```
MEMORY
{
/* Code Start branch to _c_int00 */
BEGIN : origin = 0x80000, length = 0x0002
/* User calculated golden CMAC tag for Flash Sector 0 */
GOLDEN_CMAC_TAG : origin = 0x80002, length = 0x0008
/* Flash Sector 0 containing application code */
FLASH_SECTOR_0 : origin = 0x8000A, length = 0x1FF6
.
.
.
}
```

4.7.5 Memory Maps

4.7.5.1 Boot ROM Memory Maps

Table 4-25 details the ROM memory map including secure and unsecure ROM.

Table 4-25. Boot ROM Memory Map

Memory	Origin Address	Length (Words)
Zone 1 Secure Flash Boot	0x003F 8010	0x0900
Zone 1 CRC Code	0x003F 8910	0x0600
ROM Signature	0x003F 9000	0x0002
Version	0x003F 9002	0x0004
IQmath Tables	0x003F 9006	0x1674
FPU32 Fast Tables	0x003F A67A	0x081A
FPU32 Twiddle Tables	0x003F AE94	0x0DF8
SW CRC Table	0x003F BC8C	0x0200
Boot Code	0x003F BE8C	0x2500
Interrupt Handlers	0x003F FDB4	0x0142
Flash API Table	0x003F E98C	0x0028
AES Tables	0x003F E9B4	0x1400
RTS LIB	0x003F FEF4	0x00C0
Vector Table	0x003F FFBE	0x0042

4.7.5.2 Reserved RAM Memory Maps

Table 4-26 details memory usage in RAM that is reserved for boot ROM to use. These memory sections can be reserved in the user application.

Table 4-26. Reserved RAM Memory Map

Memory	Description	Origin Address	Length (Words)
RAM	Boot Status, Boot Mode, Boot Stack	0x0000 0002	0x01BE

4.7.6 ROM Tables

Table 4-27 details the boot ROM symbol libraries that can be integrated into an application to use the available ROM functions and tables.

Table 4-27. ROM Symbol Tables

ROM Symbols	Library Name	Location
ROM Bootloaders and Functions	F280013xCPU_BootROM_Symbols	Under */libraries/boot_rom* in C2000Ware
FPU32 Tables	F280013xCPU_BootROM_Symbols	
IQmath	F280013xCPU_IQMathROM_Symbols	

4.7.7 Boot Modes and Loaders

The available boot modes and bootloaders supported on this device are detailed in this section.

4.7.7.1 Boot Modes

This section details the available boot modes that do not involve a peripheral boot loader. Table 4-28 details the available boot modes that do not involve a peripheral boot loader.

Table 4-28. Boot Mode Availability

Boot Mode	CPU Support
Flash Boot	C28x CPU
RAM Boot	C28x CPU
Wait Boot	C28x CPU
Secure Flash Boot	C28x CPU

4.7.7.1.1 Flash Boot

Flash boot mode branches to the configured memory address in Flash memory. Refer to Section 4.7.2 for all the available Flash address options.

4.7.7.1.2 RAM Boot

RAM boot mode branches to the configured memory address in RAM. Refer to Section 4.7.2 for all the available RAM address options.

4.7.7.1.3 Wait Boot

Wait boot mode branches to the memory address as mentioned in Section 4.7.3.

4.7.7.2 Bootloaders

This section details the available boot modes that use a peripheral boot loader. For more specific details on the supported data stream structure used by the following bootloaders, refer to Section 4.8.1.

4.7.7.2.1 SCI Boot Mode

The SCI boot mode asynchronously transfers code from SCI-A to internal memory. This boot mode only supports an incoming 8-bit data stream and follows the data flow as shown in Figure 4-4.

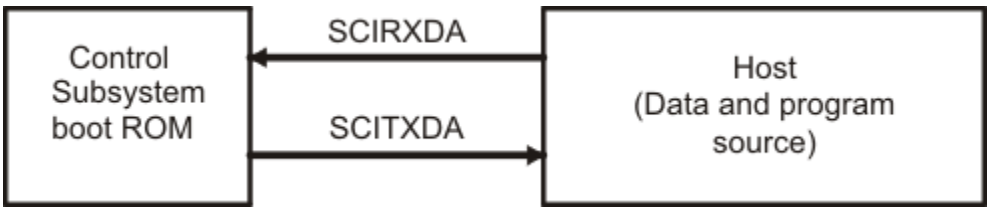


Figure 4-4. Overview of SCI Bootloader Operation

The device communicates with the external host by communication through the SCI-A peripheral. The autobaud feature of the SCI port is used to lock baud rates with the host. For this reason the SCI loader is very flexible and you can use a number of different baud rates to communicate with the device.

After each data transfer, the bootloader echoes back the 8-bit character received to the host. This allows the host to check that each character was received by the bootloader.

At higher baud rates, the slew rate of the incoming data bits can be affected by transceiver and connector performance. While normal serial communications can work well, this slew rate can limit reliable auto-baud detection at higher baud rates (typically beyond 100kbaud) and cause the auto-baud lock feature to fail. To avoid this, the following is recommended:

1. Achieve a baud-lock between the host and SCI bootloader using a lower baud rate.
2. Load the incoming application or custom loader at this lower baud rate.
3. The host can then handshake with the loaded application to set the SCI baud rate register to the desired high baud rate.

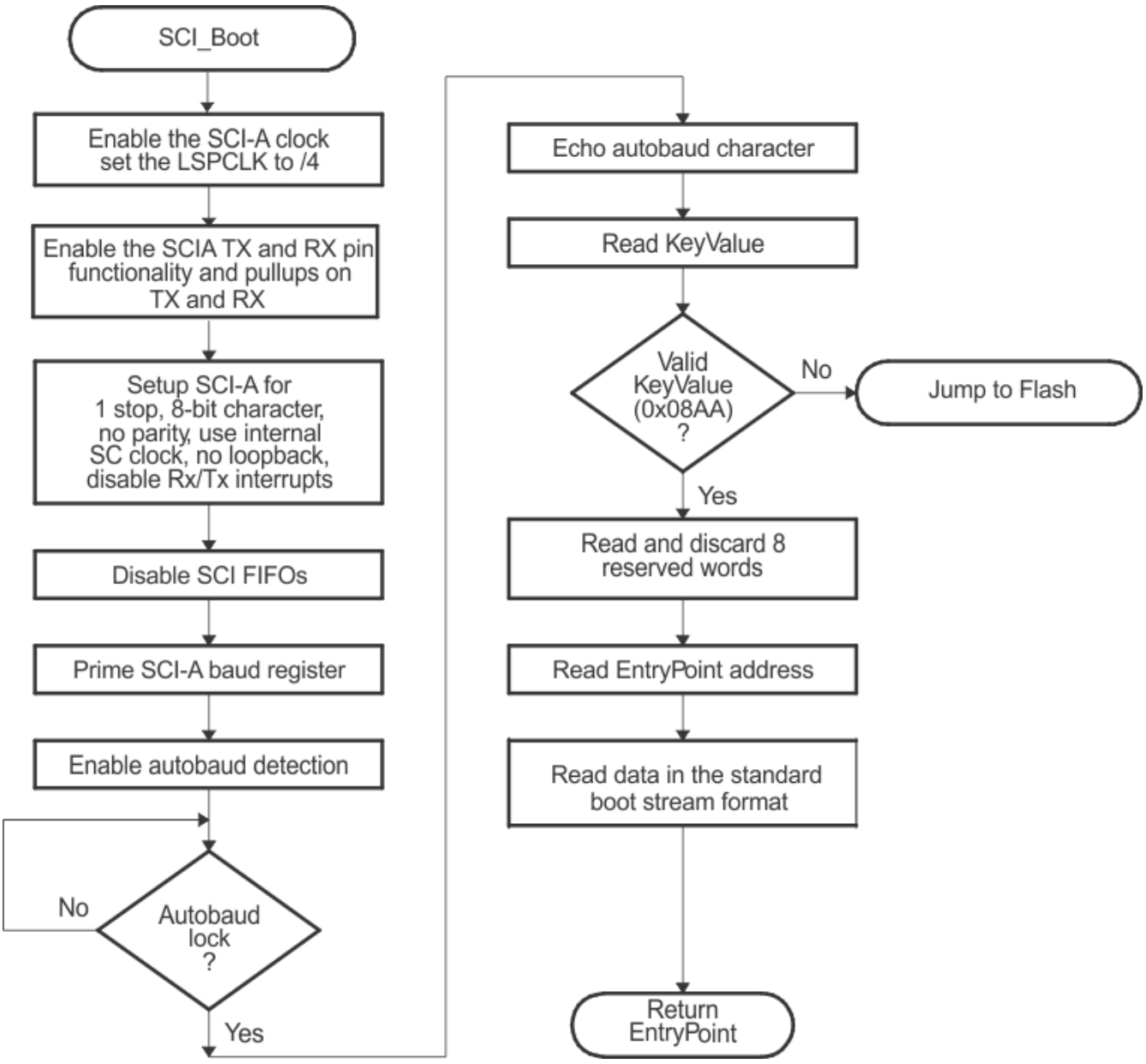


Figure 4-5. Overview of SCI Boot Function

4.7.7.2.2 SPI Boot Mode

The SPI loader expects an SPI-compatible 16-bit or 24-bit addressable serial EEPROM or serial Flash device to be present on the SPI-A pins as shown in Figure 4-6. The SPI bootloader supports an 8-bit data stream and does not support a 16-bit data stream.

Figure 4-6. Overview of SPI Bootloader Operation

The SPI boot ROM loader initializes the SPI module to interface to a serial SPI EEPROM or Flash. Devices of this type include, but are not limited to, the Xicor X25320 (4Kx8) and Xicor X25256 (32Kx8) SPI serial SPI EEPROMs and the Atmel AT25F1024A serial Flash.

The SPI boot ROM loader initializes the SPI with the following settings: FIFO enabled, 8-bit character, internal SPICLK controller mode and talk mode, clock phase = 1, polarity = 0, using the slowest baud rate.

If the download is to be performed from an SPI port on another device, then that device must be set up to operate in the peripheral mode and mimic a serial SPI EEPROM. Immediately after entering the SPI_Boot function, the pin functions for the SPI pins are set to primary and the SPI is initialized. The initialization is done at the slowest speed possible. Once the SPI is initialized and the key value read, you can specify a change in baud rate or low speed peripheral clock. Table 4-29 shows the 8-bit data stream used by the SPI.

Table 4-29. SPI 8-Bit Data Stream

Byte	Contents
1	LSB: AA (KeyValue for memory width = 8-bits)
2	MSB: 08h (KeyValue for memory width = 8-bits)
3	LSB: LOSPCP
4	MSB: SPIBRR
5	LSB: reserved for future use
6	MSB: reserved for future use
...	...
...	Reserved
...	...
17	LSB: reserved for future use
18	MSB: reserved for future use
19	LSB: Upper half (MSW) of Entry point PC[23:16]
20	MSB: Upper half (MSW) of Entry point PC[31:24] (Note: Always 0x00)
21	LSB: Lower half (LSW) of Entry point PC[7:0]
22	MSB: Lower half (LSW) of Entry point PC[15:8]
...	....
...	Data for this section.
...	...
...	Blocks of data in the format size/destination address/data as shown in the generic data stream description
...	...
...	Data for this section.
n	LSB: 00h

Table 4-29. SPI 8-Bit Data Stream (continued)

Byte	Contents
n+1	MSB: 00h - indicates the end of the source

The data transfer is done in "burst" mode from the serial SPI EEPROM. The transfer is carried out entirely in byte mode (SPI at 8 bits/character). A step-by-step description of the sequence follows:

1. The SPI-A port is initialized.
2. The GPIO pin, as defined by SPI option configured from Table 4-34, is used as a chip-select for the serial SPI EEPROM or Flash.
3. The SPI-A outputs a read command for the serial SPI EEPROM or Flash.
4. The SPI-A sends the serial SPI EEPROM an address 0x0000; that is, the host requires that the EEPROM or Flash must have the downloadable packet starting at address 0x0000 in the EEPROM or Flash. The loader is compatible with both 16-bit addresses and 24-bit addresses.
5. The next word fetched must match the key value for an 8-bit data stream (0x08AA). The least significant byte of this word is the byte read first and the most significant byte is the next byte fetched. This is true of all word transfers on the SPI. If the key value does not match, then the load is aborted and the bootloader jumps to Flash.
6. The next two bytes fetched can be used to change the value of the low speed peripheral clock register (LOSPCP) and the SPI baud rate register (SPIBRR). The first byte read is the LOSPCP value and the second byte read is the SPIBRR value. The next seven words are reserved for future enhancements. The SPI bootloader reads these seven words and discards them.
7. The next two words makeup the 32-bit entry point address where execution continues after the boot load process is complete. This is typically the entry point for the program being downloaded through the SPI port.
8. Multiple blocks of code and data are then copied into memory from the external serial SPI EEPROM through the SPI port. The blocks of code are organized in the standard data stream structure presented earlier. This is done until a block size of 0x0000 is encountered. At that point in time the entry point address is returned to the calling routine that then exits the bootloader and resumes execution at the address specified.

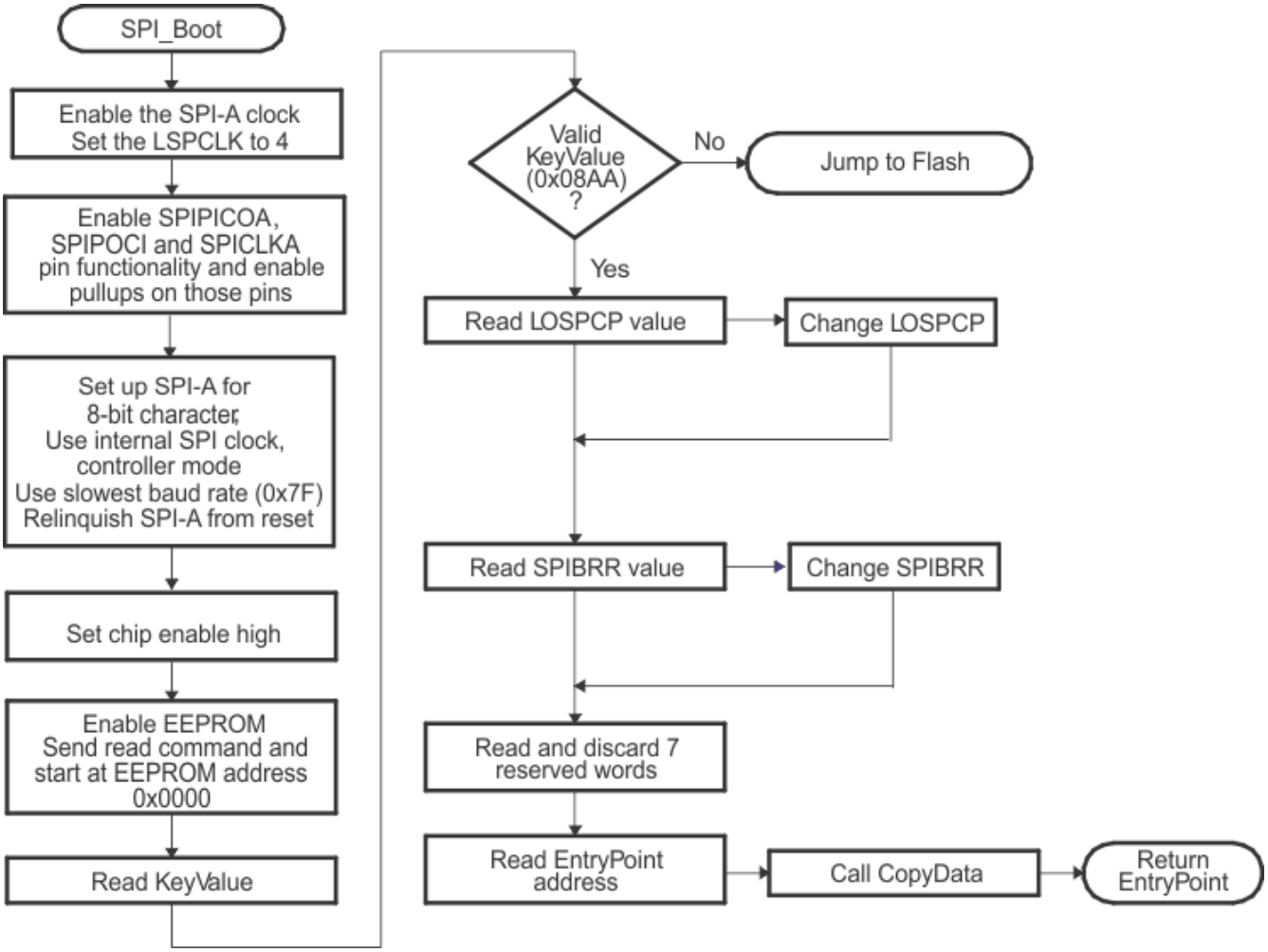


Figure 4-7. Data Transfer from EEPROM Flow

4.7.7.2.3 I2C Boot Mode

The I2C bootloader expects an 8-bit wide I2C-compatible EEPROM device to be present at address 0x50 on the I2C-A bus as shown in Figure 4-8. The EEPROM must adhere to conventional I2C EEPROM protocol, as described in this section, with a 16-bit base address architecture.

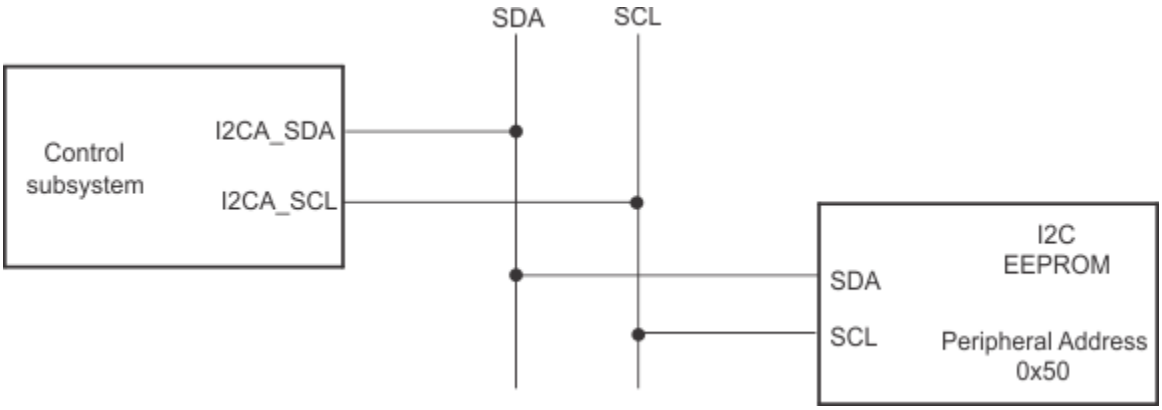


Figure 4-8. EEPROM Device at Address 0x50

If the download is to be performed from a device other than an EEPROM, then that device must be set up to operate in the target mode and mimic the I2C EEPROM. Immediately after entering the I2C boot function, the GPIO pins are configured for I2C-A operation and the I2C is initialized. The following requirements must be met when booting from the I2C module:

- The input frequency to the device must be in the appropriate range.
- The EEPROM must be at target address 0x50.

The bit-period prescalers (I2CCLKH and I2CCLKL) are configured by the bootloader to run the I2C at a 50 percent duty cycle at 100kHz bit rate (standard I2C mode) when the system clock is 8MHz. These registers can be modified after receiving the first few bytes from the EEPROM. This allows the communication to be increased up to a 400kHz bit rate (fast I2C mode) during the remaining data reads.

Arbitration, bus busy, and target signals are not checked. Therefore, no other controller is allowed to control the bus during this initialization phase. If the application requires another controller during I2C boot mode, that controller must be configured to hold off sending any I2C messages until the application software signals that the software is past the bootloader portion of initialization.

The non-acknowledgment bit is checked only during the first message sent to initialize the EEPROM base address. This is to make sure that an EEPROM is present at address 0x50 before continuing. If an EEPROM is not present, the non-acknowledgment bit is not checked during the address phase of the data read messages (I2C_Get Word). If a non-acknowledgment is received during the data read messages, the I2C bus hangs. Table 4-30 shows the 8-bit data stream used by the I2C.

The I2C EEPROM protocol required by the I2C bootloader is shown in Figure 4-10 and Figure 4-11. The first communication, which sets the EEPROM address pointer to 0x0000 and reads the KeyValue (0x08AA), is shown in Figure 4-10. All subsequent reads are shown in Figure 4-11 and are read two bytes at a time.

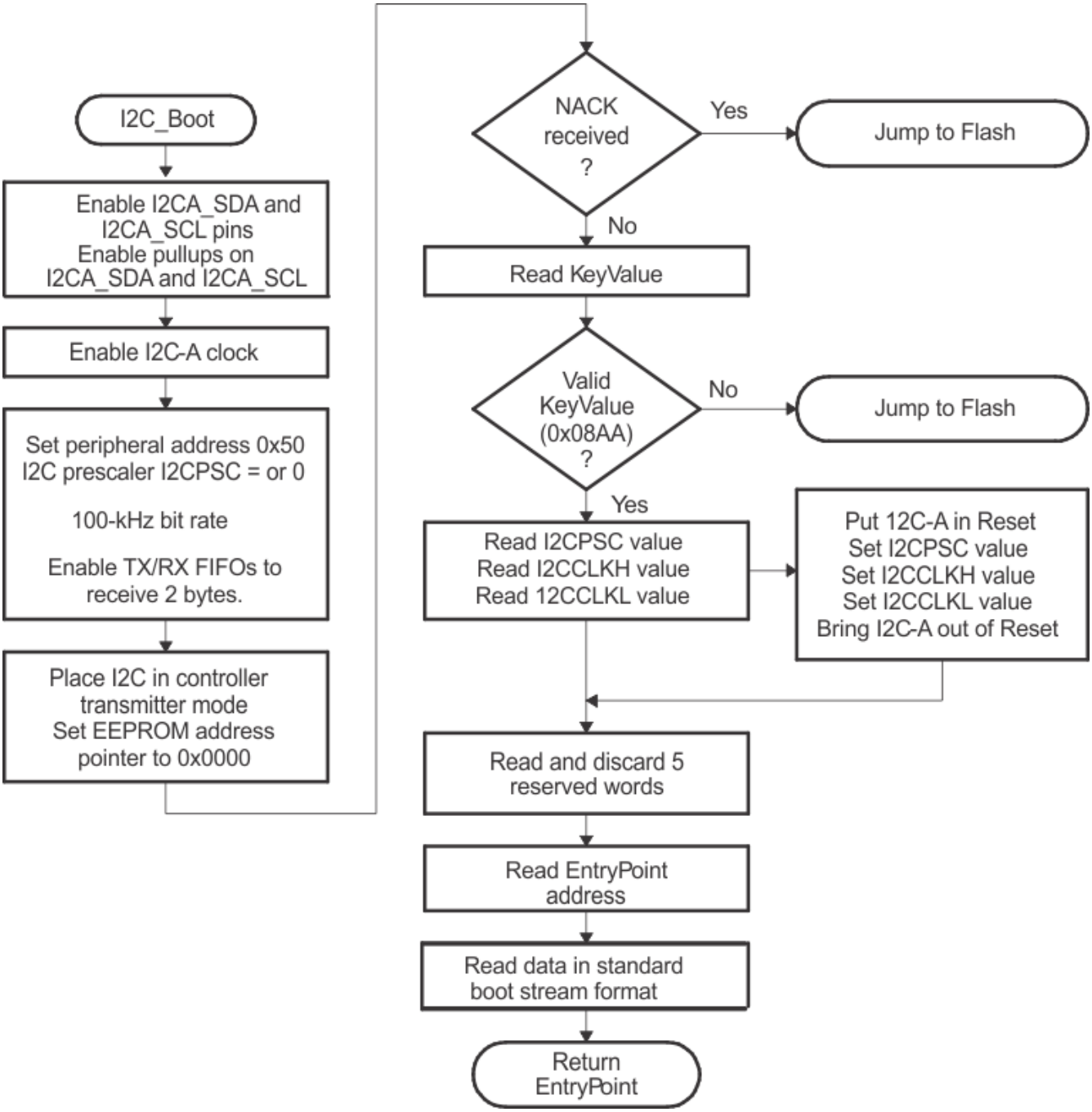


Figure 4-9. Overview of I2C Boot Function

Table 4-30. I2C 8-Bit Data Stream

Byte	Contents
1	LSB: AA (KeyValue for memory width = 8 bits)
2	MSB: 08h (KeyValue for memory width = 8 bits)
3	LSB: I2CPSC[7:0]
4	Reserved
5	LSB: I2CCLKH[7:0]
6	MSB: I2CCLKH[15:8]
7	LSB: I2CCLKL[7:0]
8	MSB: I2CCLKL[15:8]
...	...
...	Data for this section.
...	...
17	LSB: Reserved for future use
18	MSB: Reserved for future use
19	LSB: Upper half of entry point PC
20	MSB: Upper half of entry point PC[22:16] (Note: Always 0x00)
21	LSB: Lower half of entry point PC[15:8]
22	MSB: Lower half of entry point PC[7:0]
...	...
...	Data for this section.
...	...
...	Blocks of data in the format size/destination address/data as shown in the generic data stream description.
...	...
...	Data for this section.
n	LSB: 00h
n+1	MSB: 00h - indicates the end of the source

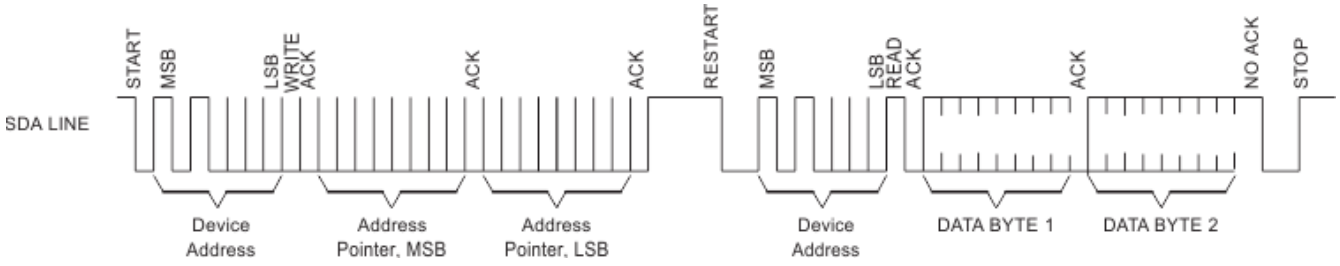


Figure 4-10. Random Read

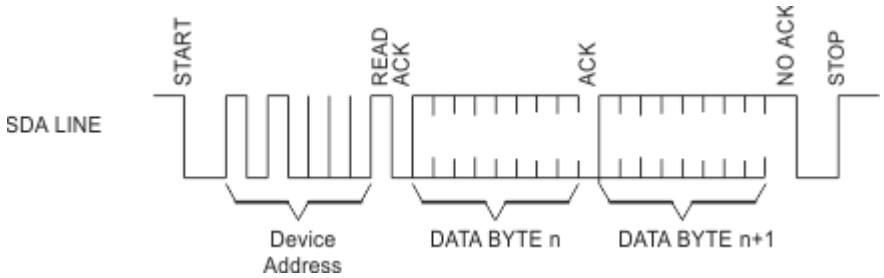


Figure 4-11. Sequential Read

4.7.7.2.4 Parallel Boot Mode

The parallel general-purpose I/O (GPIO) boot mode asynchronously transfers code from host to C28x internal memory. Each value is 8-bits long and follows the same data flow as outlined in Figure 4-12.

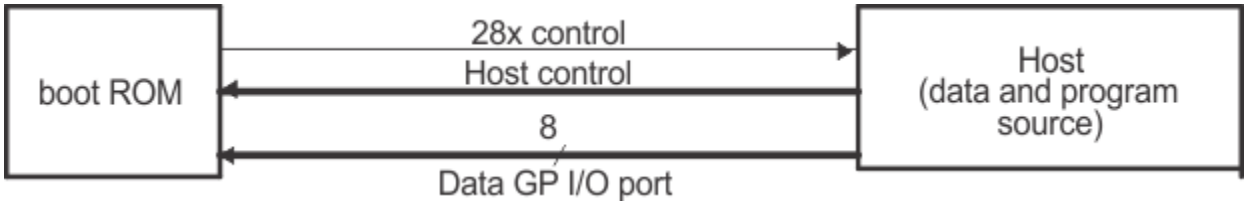


Figure 4-12. Overview of Parallel GPIO Bootloader Operation

The control subsystem communicates with the external host device by polling/driving the Host Control and 28x control lines. The handshake protocol shown in Figure 4-13 must be used to successfully transfer each word using GPIO [D0:D7]. This protocol is very robust and allows for a slower or faster host to communicate with the controller subsystem.

Two consecutive 8-bit words are read to form a single 16-bit word. The least-significant byte (LSB) is read first followed by the most-significant byte (MSB). In this case, data is read from GPIO[D0:D7].

The 8-bit data stream is shown in Table 4-31.

Table 4-31. Parallel GPIO Boot 8-Bit Data Stream

Bytes		GPIO[D0:D7] (Byte 1 of 2)	GPIO[D0:D7] (Byte 2 of 2)	Description
1	2	AA	08	0x08AA (KeyValue for memory width = 16 bits)
3	4	00	00	8 reserved words (words 2 - 9)
...	...	...	...	...
17	18	00	00	Last reserved word
19	20	BB	00	Entry point PC[22:16]
21	22	DD	CC	Entry point PC[15:0] (PC = 0x00BBCCDD)
23	24	NN	MM	Block size of the first block of data to load = 0xMMNN words
25	26	BB	AA	Destination address of first block Addr[31:16]
27	28	DD	CC	Destination address of first block Addr[15:0] (Addr = 0xAABBCCDD)
29	30	BB	AA	First word of the first block in the source being loaded = 0xAABB
... ...				... Data for this section. ...
.		BB	AA	Last word of the first block of the source being loaded = 0xAABB
.		NN	MM	Block size of the 2nd block to load = 0xMMNN words
.		BB	AA	Destination address of second block Addr[31:16]
.		DD	CC	Destination address of second block Addr[15:0]
.		BB	AA	First word of the second block in the source being loaded
.				…
n	n+1	BB	AA	Last word of the last block of the source being loaded (More sections if required)
n+2	n+3	00	00	Block size of 0000h - indicates end of the source program

The device first signals the host that the device is ready to begin data transfer by pulling the 28x control pin low. The host load then initiates the data transfer by pulling the DSP control pin low. The complete protocol is shown in Figure 4-13.

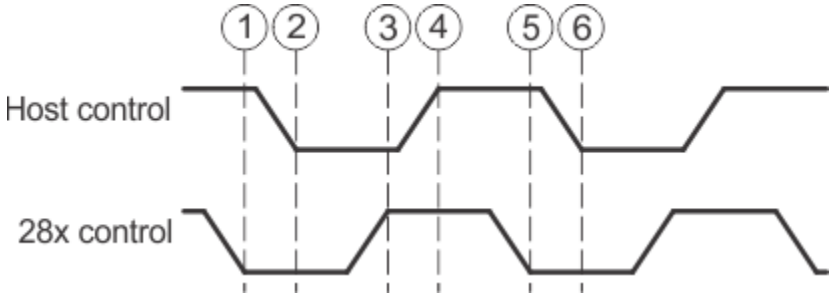


Figure 4-13. Parallel GPIO Bootloader Handshake Protocol

1. The device indicates the device is ready to start receiving data by pulling the 28x control pin low.
2. The bootloader waits until the host puts data on GPIO [D0:D7]. The host signals to the device that data is ready by pulling the host control pin low.
3. The device reads the data and signals the host that the read is complete by pulling the 28x control pin high.
4. The bootloader waits until the host acknowledges the device by pulling the host control pin high.
5. The device again indicates the device is ready for more data by pulling the 28x control pin low.

This process is repeated for each data value to be sent.

Figure 4-14 shows an overview of the Parallel GPIO bootloader flow.

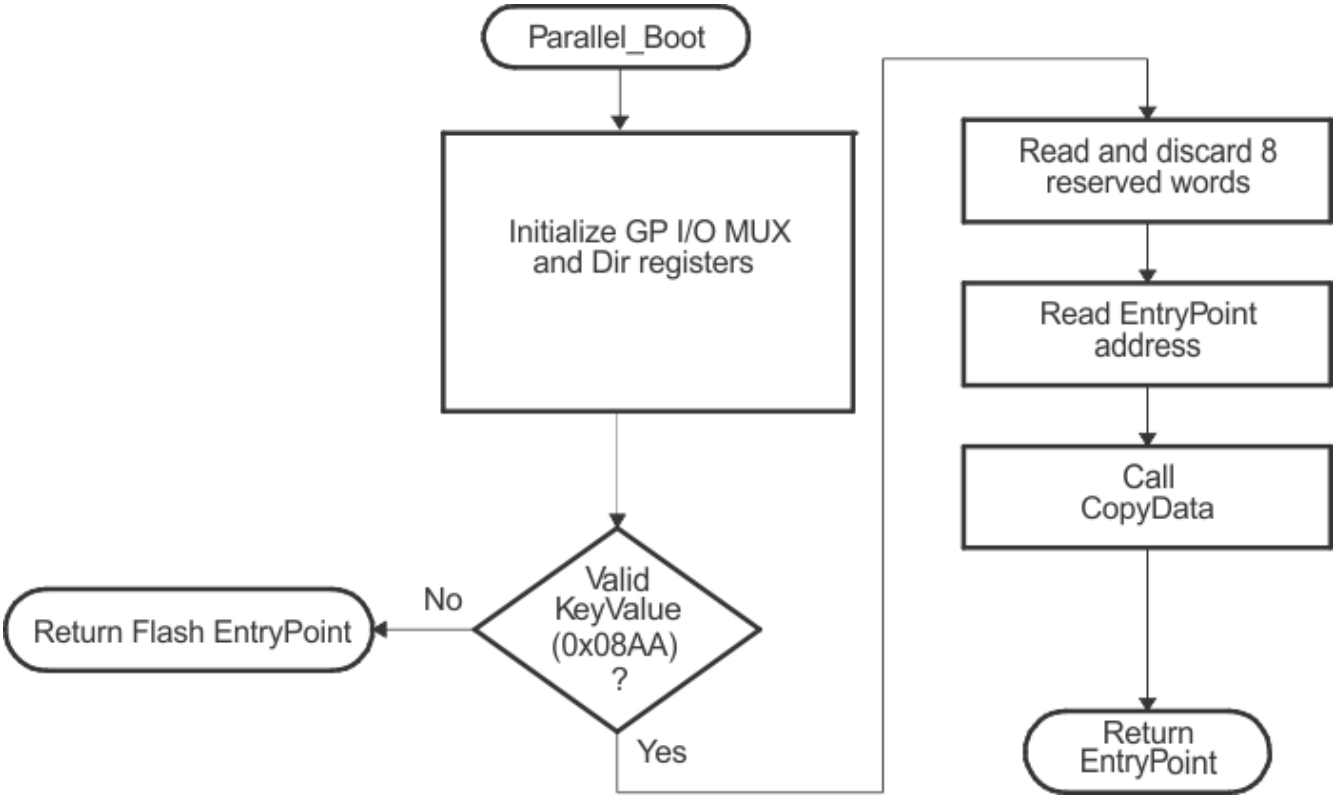


Figure 4-14. Overview of Parallel GPIO Boot Function

Figure 4-15 shows the transfer flow from the host side. The operating speed of the CPU and host are not critical in this mode as the host waits for the device and the device in turn waits for the host. In this manner, the protocol works with both a host running faster and a host running slower than the device.

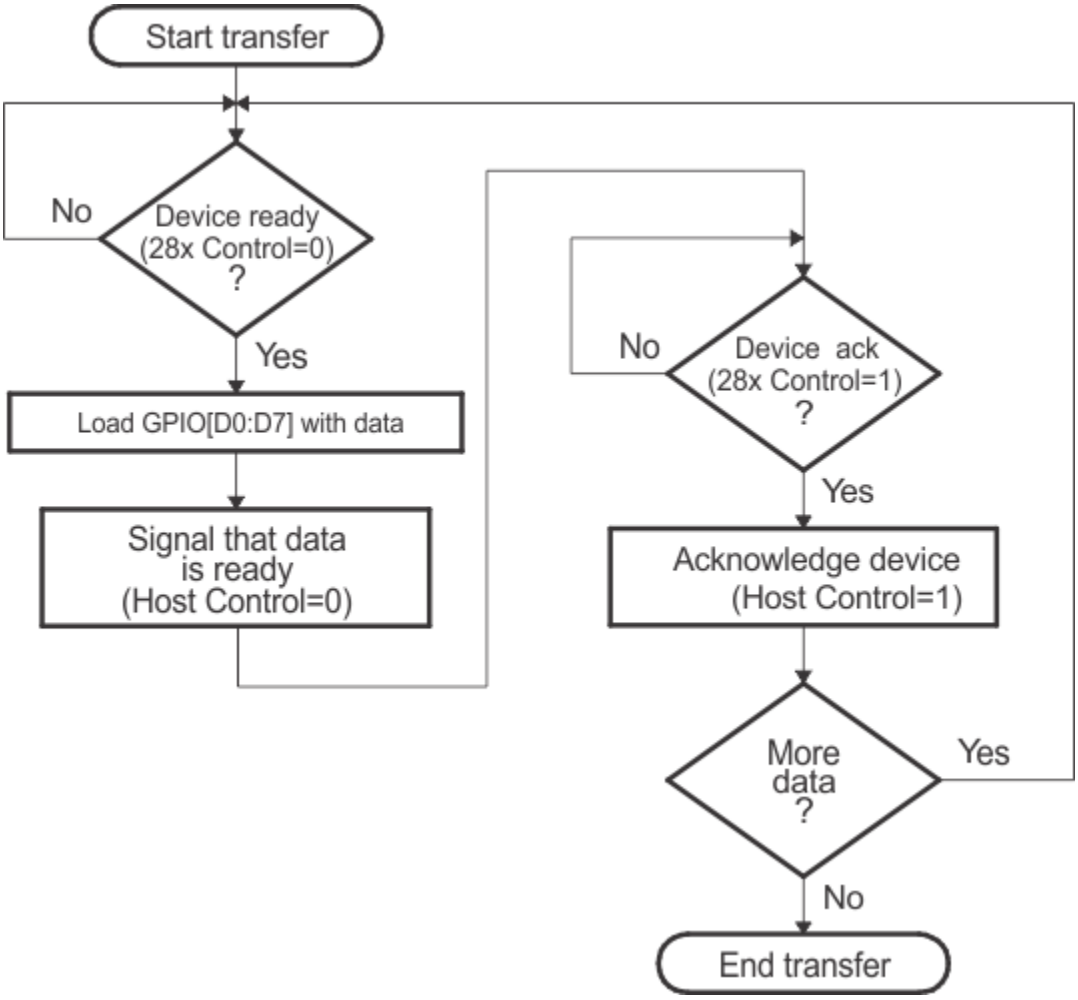


Figure 4-15. Parallel GPIO Mode - Host Transfer Flow

Figure 4-16 shows the flow used to read a single word of data from the parallel port. The 8-bit routine, shown in Figure 4-16, discards the upper 8 bits of the first read from the port and treats the lower 8 bits masked with D7 in bit position 7 and D6 in bit position 6 as the least-significant byte (LSB) of the word to be fetched. The routine then performs a second read to fetch the most-significant byte (MSB). The routine then performs a second read to fetch the most-significant byte (MSB). The routine then combines the MSB and LSB into a single 16-bit value to be passed back to the calling routine.

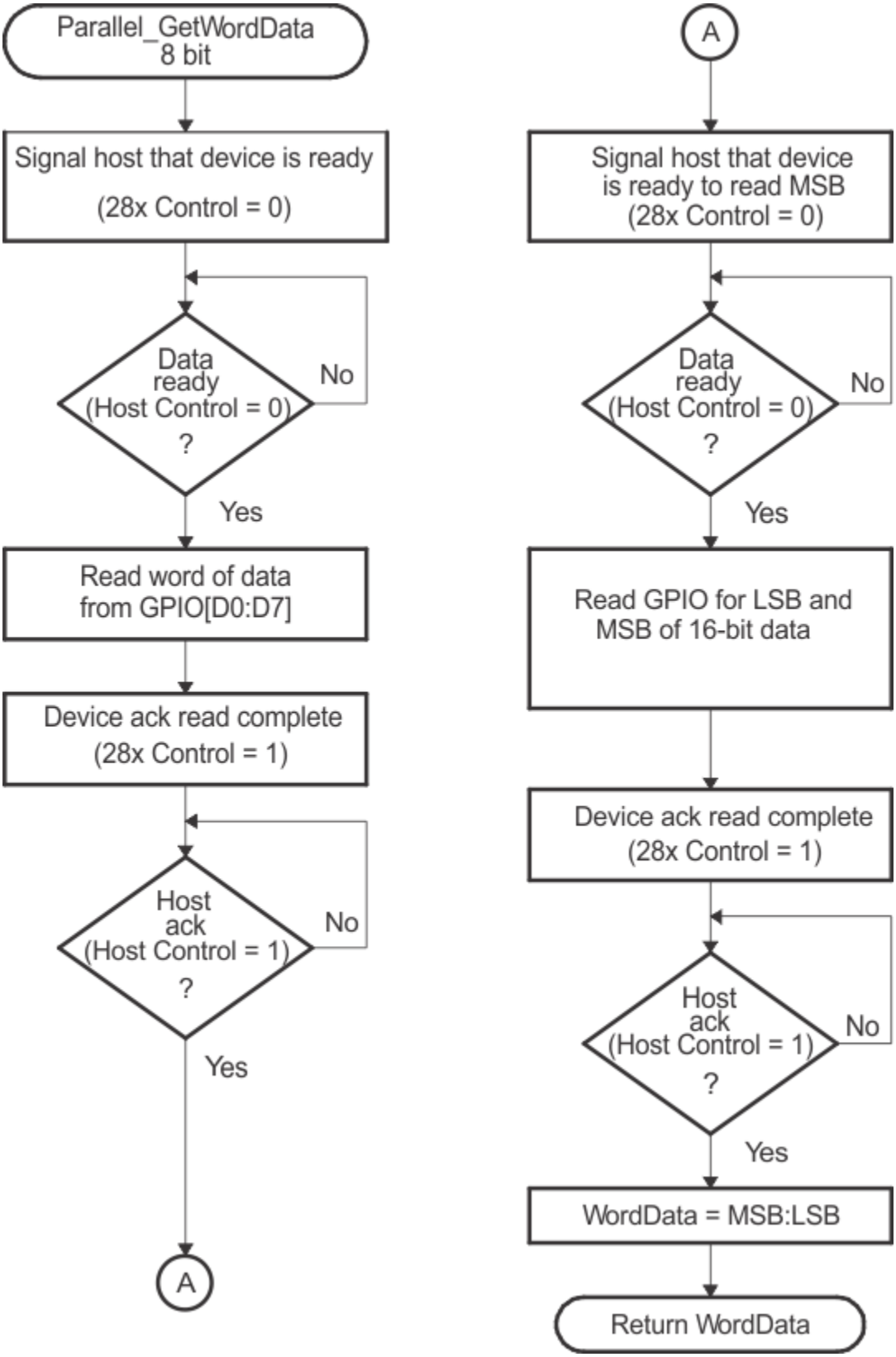


Figure 4-16. 8-Bit Parallel GetWord Function

4.7.8 GPIO Assignments

This section details the GPIOs and boot option values used for boot mode set in the BOOT_DEF memory location located at Z1-OTP-BOOTDEF-LOW/ Z2-OTP-BOOTDEF-LOW and Z1-OTP-BOOTDEF-HIGH/ Z2-OTP-BOOTDEF-HIGH. Refer to Section 4.4.2 on how to configure BOOT_DEFx. When selecting a boot mode option, make sure to verify that the necessary pins are available in the pin mux options for the specific device package being used.

Default boot mode GPIO pins:

- Boot mode pin 0 - GPIO32
- Boot mode pin 1 - GPIO24

Guidelines on boot pin selection:

- Avoid pins that have PWM functionality.
- Can not be analog or USB pins.
- Boot mode select pins and default boot peripheral pins can be available on all packages.
- Avoid JTAG emulation pins and crystal pins.
- Boot mode select pins can be inputs.
- Pins can not have PHY bootstrap functionality.

Table 4-32. SCI Boot Options

Option	BOOTDEFx Value	SCITXDA GPIO	SCIRXDA GPIO	Package Supported
0 (default)	0x01	GPIO29	GPIO28	All
1	0x21	GPIO1	GPIO0	All
3	0x61	GPIO7	GPIO3	48-PT, 32-RHB, 32-VFC
4	0x81	GPIO16	GPIO3	48-PT

Table 4-33. I2C Boot Options

Option	BOOTDEFx Value	SDAA GPIO	SCLA GPIO	Package Supported
0	0x07	GPIO0	GPIO1	All
1	0x27	GPIO32	GPIO33	48-PT
2	0x47	GPIO5	GPIO4	All

Table 4-34. SPI Boot Options

Option	BOOTDEFx Value	SPIPICOA	SPIPOCIA	SPICLKA	SPIPTEA	Package Supported
0	0x06	GPIO24	GPIO1	GPIO3	GPIO5	All
1	0x26	GPIO16	GPIO1	GPIO3	GPIO0	48-PT
3	0x66	GPIO16	GPIO13	GPIO12	GPIO24	48-PT

Table 4-35. Parallel Boot Options

Option	BOOTDEFx Value	D0-D7 GPIO	DSP Control GPIO	Host Control GPIO	Package Supported
0 (default)	0x00	D0-D7 (GPIO 0,1,3,4,5,24,28,29)	GPIO224	GPIO242	All
1	0x20	D0-D7 (GPIO 0-7)	GPIO12	GPIO13	48-PT

4.7.9 Secure ROM Function APIs

There are a few functions that are available within Secure ROM to be called by the application to perform EXEONLY Flash/RAM tasks in a secure manner.

Note

The application can disable interrupts before calling one of the EXEONLY function APIs.

If a vector fetch request is given by the CPU (C28x) while the program counter (PC) is within the EXEONLY function API code of the Secure ROM, a reset fires (RSN if from C28x). The consequence of this is if an NMI or ITRAP or Bus Fault occurs while the PC is executing one of the EXEONLY API functions, the NMI/ITRAP/Fault cannot be serviced because a reset is fired to the subsystem.

The secure CRC calculation zone 1 function allows a safety CRC check of EXEONLY memory in a secure manner. The CRC length provided must be a value from 1 to 8 where 1 represents a CRC size of 32 16-bit words and 8 represents a CRC size of 4096 16-bit words. The source address specifies the starting address for the CRC and the golden CRC specifies what the calculated CRC is expected to be. The source and destination memories must be configured for the same zone (Z1 is only compared to Z1, and Z2 is only compared to Z2). Additionally, the CRC length must not cross over the Flash sector or RAM block boundary. Any violations of these requirements results in a failure status returned. Upon successful CRC, the number of 16-bit words CRCed or an error code is returned.

Table 4-36. Secure CRC Calculation Function

CPU	Function Prototype	Function Parameters	Function Return Value
CPU (C28x)	uint16_t **SecureCRCCalcZ1**(uint32_t goldenCRC, uint16_t len_id, uint_16_t *src)	*len_id* : A number from 1 to 8 that corresponds to length options of 32, 64, 128, 256, 512, 1024, 2048, or 4096 16-bit words *goldenCRC* : The golden CRC that the calculated CRC is expected to match *src* : The source memory address to begin CRC calculation	0xXXXX : Returns the number of 16-bit words CRCed 0xFF00 : CRC Mismatch 0x0FF01 : Invalid length option 0xFF02 : Source address is not modulo of length value 0xFF03 : CRC size crosses over Flash sector or RAM block boundary 0xFF04 : The source and destination memory do not belong to the same zone.

The CMAC calculate and compare function allows to calculate CMAC signature of a Flash memory block and compare against a golden signature. This is used in the secure boot mode to authenticate the boot image.

Table 4-37. CMAC Calculation Function

CPU	Function Prototype	Function Parameters	Function Return Value
CPU (C28x)	uint32_t **CPU1BROM_calculateCMAC**(uint32_t startAddress, uint32_t endAddress, uint32_t signatureAddress)	**startAddress**: Starting address of memory for which CMAC has to be calculated **endAddress**: Ending address of memory for which CMAC has to be calculated **signatureAddress**: Address of location where golden CMAC signature is stored	**0xFFFF FFFFU**: Calculated CMAC signature did not match golden signature (fail) **0xA5A5 A5A5U**: Memory range provided is not aligned to 128-bit boundary or length is zero **0xE1E1 E1E1U**: AES Engine timed out **0x0000 0000U**: No Error

4.7.10 Clock Initializations

During boot-up, the boot ROM initializes the device clocking, depending upon the reset source, to assist in faster boot time response. Clock configurations are performed by the boot ROM code only for POR and XRS reset types. For all other resets, the boot ROM starts executing with the clocks that were already set up before reset.

Note

CPU performs clock configurations during boot up. If the PLL is used during the boot process, the PLL is bypassed by the boot ROM code before branching to the user application.

Table 4-38. CPU Boot Clock Sources

Source	Frequency	Description
WROSCDIV8	2.5MHz to 8.75MHz	Default clock source
SYSOSCDIV4	8MHz	Alternate clock source
HPLL	160MHz	Boot ROM configures to use PLL output clock. This can be skipped (bootROM to use SYSOSC) by configuring TI OTP.

Table 4-39. CPU Clock State After Boot

Reset Source	Clock State
POR/XRS	1. Use SYSOSC 2. Modify the SYSDIVSEL based on TRIM (TI OTP) configuration (minimum divider is /1). 3. Power up PLL and set an integer multiplier. Check PLL Lock status and put PLL output in clock path – This is gated by OTP config as well to make sure successful boot on fresh parts on bench. 4. After PLL output is enabled in clock path, CPU is executing at PLLOUT/ SYSCLKDIV (can be configured using TRIM (TI OTP)) frequency.
All other Resets	Maintain clocks setup before device reset.

4.7.11 Boot Status Information

Boot ROM keeps a record of the various actions and events that occur during boot ROM execution. The reason for this is because NMI and other exceptions are enabled by default in the device and must be handled accordingly. Boot ROM stores the boot status information in a RAM location so that the user application can read this boot status and take the necessary actions per application's needs to handle these events.

4.7.11.1 Booting Status

Boot ROM health and booting status is written to a 32-bit address in M0RAM. This status is cleared on a POR or XRS reset. The previous status is retained on any other reset. For example, clear the status before performing a debugger device reset to view the latest boot ROM actions reflected in the status.

Table 4-40. Boot Status Address

Description	Address
Boot ROM Status	0x0000 0002

Table 4-41. Boot Status Bit Fields

Value	Description
0x4000 0000	Flash 2T Not Ready
0x2000 0000	TRIM Load Error
0x1000 0000	RAM Init Error
0x0800 0000	Flash Verification Error
0x0200 0000	DCSM Initialization Invalid LP
0x0100 0000	SYSPLL enabled successfully
0x0040 0000	Missing clock NMI occurred
0x0010 0000	Memory Uncorrectable Error NMI occurred
0x0008 0000	RL NMI occurred
0x0004 0000	ERAD NMI occurred
0x0002 0000	Boot ROM detected a PIE mismatch
0x0001 0000	Boot ROM detected an ITRAP
0x0000 8000	Boot ROM has completed running
0x0000 4000	Watchdog self test fail
0x0000 2000	Boot ROM handled POR
0x0000 1000	Boot ROM handled XRS
0x0000 0800	Boot ROM handled all the resets
0x0000 0200	DCSM initialization has completed
0x0000 0100	RAM Initialization Complete
0x0000 000B	Wait boot has started
0x0000 0008	I2C boot has started
0x0000 0007	SPI boot has started
0x0000 0006	SCI boot has started
0x0000 0005	RAM boot has started
0x0000 0004	Parallel boot has started
0x0000 0003	Secure Flash boot has started
0x0000 0002	Flash boot has started
0x0000 0001	Boot ROM has started running

4.7.12 ROM Version

The ROM revision and release date information is stored at the ROM locations specified in Table 4-42. Reading a revision number value of “0x100” represents version “1.0”, “0x101” represents version “1.1”, and so on. Reading a revision date value of “0x0119” represents “01/19” or “January 2019”.

Table 4-42. Boot ROM Version Information

Contents	Address
Revision Number	0x003F 9002
Revision Date	0x003F 9803
Build Number	0x003F 9804

4.8 Application Notes for Using the Bootloaders

4.8.1 Bootloader Data Stream Structure

This section details the data transfer protocols or stream structures that allow boot data transfer between boot ROM and host device. This data transfer protocol is compatible to the respective bootloaders on C2000 devices.

Table 4-43 and Example 4-2 show the structure of the data stream incoming to the bootloader. The basic structure is the same for all the bootloaders and is based on the C54x source data stream generated by the C54x hex utility. The C28x hex utility (hex2000.exe) has been updated to support this structure. The hex2000.exe utility is included with the C2000 code generation tools. All values in the data stream structure are in hex. Refer to Section 4.8.2 for more details on using the C28x hex utility to convert a project to this format.

The first 16-bit word in the data stream is known as the key value. The key value is used to indicate to the bootloader the width of the incoming stream: 8 or 16 bits. Note that not all bootloaders accept both 8- and 16-bit streams. Refer to the detailed information on each loader for the valid data stream width. For an 8-bit data stream, the key value is 0x08AA; for a 16-bit data stream, the key value is 0x10AA. If a bootloader receives an invalid key value, then the load is aborted.

The next eight words are used to initialize register values or otherwise enhance the bootloader by passing values to the bootloader. If a bootloader does not use these values then the values are reserved for future use and the bootloader simply reads the value and then discards the value. Currently only the SPI and I2C and parallel bootloaders use these words to initialize registers.

The tenth and eleventh words comprise the 22-bit entry point address. This address is used to initialize the PC after the boot load is complete. This address is most likely the entry point of the program downloaded by the bootloader.

The twelfth word in the data stream is the size of the first data block to be transferred. The size of the block is defined as 8-bit data stream format. For example, to transfer a block of 20 8-bit data values from an 8-bit data stream, the block size is 0x000A to indicate 10 16-bit words.

The next two words indicate to the loader the destination address of the block of data. Following the size and address is the 16-bit words that makeup that block of data.

This pattern of block size/destination address repeats for each block of data to be transferred. Once all the blocks have been transferred, a block size of 0x0000 signals to the loader that the transfer is complete. At this point, the loader returns the entry point address to the calling routine, which cleans up and exits. Execution then continues at the entry point address as determined by the input data stream contents.

Table 4-43. LSB/MSB Loading Sequence in 8-Bit Data Stream

Byte		Contents	
		LSB (First Byte of 2)	MSB (Second Byte of 2)
1	2	LSB: AA (KeyValue for memory width = 8 bits)	MSB: 08h (KeyValue for memory width = 8 bits)
3	4	LSB: Register initialization value or reserved	MSB: Register initialization value or reserved
5	6	LSB: Register initialization value or reserved	MSB: Register initialization value or reserved
7	8	LSB: Register initialization value or reserved	MSB: Register initialization value or reserved
...	...	...	...
...	...	...	...
17	18	LSB: Register initialization value or reserved	MSB: Register initialization value or reserved
19	20	LSB: Upper half of Entry point PC[23:16]	MSB: Upper half of entry point PC[31:24] (Always 0x00)
21	22	LSB: Lower half of Entry point PC[7:0]	MSB: Lower half of Entry point PC[15:8]
23	24	LSB: Block size in words of the first block to load. If the block size is 0, this indicates the end of the source program. Otherwise another block follows. For example, a block size of 0x000A indicates 10 words or 20 bytes in the block.	MSB: block size
25	26	LSB: MSW destination address, first block Addr[23:16]	MSB: MSW destination address, first block Addr[31:24]
27	28	LSB: LSW destination address, first block Addr[7:0]	MSB: LSW destination address, first block Addr[15:8]
29	30	LSB: First word of the first block being loaded	MSB: First word of the first block being loaded
...	...	...	...
...	...	...	...
.	.	LSB: Last word of the first block to load	MSB: Last word of the first block to load
.	.	LSB: Block size of the second block	MSB: Block size of the second block
.	.	LSB: MSW destination address, second block Addr[23:16]	MSB: MSW destination address, second block Addr[31:24]
.	.	LSB: LSW destination address, second block Addr[7:0]	MSB: LSW destination address, second block Addr[15:8]
.	.	LSB: First word of the second block being loaded	MSB: First word of the second block being loaded
...	...	...	...
...	...	...	...
.	.	LSB: Last word of the second block	MSB: Last word of the second block
.	.	LSB: Block size of the last block	MSB: Block size of the last block
.	.	LSB: MSW of destination address of last block Addr[23:16]	MSB: MSW destination address, last block Addr[31:24]
.	.	LSB: LSW destination address, last block Addr[7:0]	MSB: LSW destination address, last block Addr[15:8]
.	.	LSB: First word of the last block being loaded	MSB: First word of the last block being loaded
...	...	...	...
...	...	...	...
.	.	LSB: Last word of the last block	MSB: Last word of the last block
n	n+1	LSB: 00h	MSB: 00h - indicates the end of the source

Example 4-2. Data Stream Structure 8-bit

```
AA 08       ; 0x08AA 8bit key value
00 00 00 00 ; 8 reserved words
00 00 00 00
00 00 00 00
00 00 00 00
3F 00 00 80 ; 0x003F8000 EntryAddr, starting point after boot load completes
05 00       ; 0x0005 First block consists of 5 16-bit words
3F 00 10 90 ; 0x003F9010 First block is loaded starting at 0x3F9010
01 00       ; Data loaded = 0x0001 0x0002 0x0003 0x0004 0x0005
02 00
03 00
04 00
05 00
02 00       ; 0x0002 - 2nd block consists of 2 16-bit words
3F 00 00 80 ; 0x003F8000 2nd block is loaded starting at 0x3F8000
00 77       ; Data loaded = 0x7700 0x7625
25 76
00 00       ; 0x0000 Size of 0 indicates end of data stream
After load has completed, the following memory values are initialized as follows:
Location Value
0x3F9010 0x0001
0x3F9011 0x0002
0x3F9012 0x0003
0x3F9013 0x0004
0x3F9014 0x0005
0x3F8000 0x7700
0x3F8001 0x7625
PC Begins execution at 0x3F8000
```

4.8.2 The C2000 Hex Utility

To use the features of the bootloader, you must generate a data stream and boot table as described in Section 4.8.1. The hex conversion utility tool, included with the C28x code generation tools, can generate the required data stream including the required boot table. This section describes the hex2000 utility. An example of a file conversion performed by hex2000 is described in Example 4-3.

The hex utility supports creation of the boot table required for the SCI, SPI, I2C, CAN, and parallel I/O loaders. That is, the hex utility adds the required information to the file such as the key value, reserved bits, entry point, address, block start address, block length and terminating value. The contents of the boot table vary slightly depending on the boot mode and the options selected when running the hex conversion utility. The actual file format required by the host (ASCII, binary, hex, and so on) differs from one specific application to another and some additional conversion can be required.

To build the boot table, follow these steps:

1. **Assemble or compile the code.** This creates the object files that are then used by the linker to create a single output file.
2. **Link the file.** The linker combines all of the object files into a single output file in common object file format (ELF). The specified linker command file is used by the linker to allocate the code sections to different memory blocks. Each block of the boot table data corresponds to an initialized section in the ELF file. Uninitialized sections are not converted by the hex conversion utility. The following options can be useful:
   - The linker -m option can be used to generate a map file. This map file shows all of the sections that were created, the location in memory, and the length. Check this file to make sure that the initialized sections are where you expect them.
   - The linker -w option configures the linker to show, if the linker assigned a section to a memory region automatically. For example, if you have a section in your code called .TI.ramfunc.
3. **Run the hex conversion utility.** Choose the appropriate options for the desired boot mode and run the hex conversion utility to convert the ELF file produced by the linker to a boot table.

See the *TMS320C28x Assembly Language Tools User's Guide* and the *TMS320C28x Optimizing C/C++ Compiler User's Guide* for more information on the compiling and linking process.

Table 4-44 summarizes the hex conversion utility options available for the bootloader. See the *TMS320C28x Assembly Language Tools User's Guide* for a detailed description of the hex2000 operations used to generate a boot table. Updates are made to support the I2C boot. See the Codegen release notes for the latest information.

Table 4-44. Boot Loader Options

Option	Description
-boot	Convert all sections into bootable form (use instead of a SECTIONS directive)
-sci8	Specify the source of the bootloader table as the SCI-A port, 8-bit mode
-spi8	Specify the source of the bootloader table as the SPI-A port, 8-bit mode
-gpio8	Specify the source of the bootloader table as the GPIO port, 8-bit mode
-bootorg value	Specify the source address of the bootloader table
-lospcp value	Specify the initial value for the LOSPCP register. This value is used only for the spi8 boot table format and ignored for all other formats. If the value is greater than 0x7F, the value is truncated to 0x7F.
-spibrr value	Specify the initial value for the SPIBRR register. This value is used only for the spi8 boot table format and ignored for all other formats. If the value is greater than 0x7F, the value is truncated to 0x7F.
-e value	Specify the entry point at which to begin execution after boot loading. The value can be an address or a global symbol. This value is optional. The entry point can be defined at compile time using the linker -e option to assign the entry point to a global symbol. The entry point for a C program is normally _c_int00 unless defined otherwise by the -e linker option.
-i2c8	Specify the source of the bootloader table as the I2C-A port, 8-bit
-i2cpsc value	Specify the value for the I2CPSC register. This value is loaded and takes effect after all I2C options are loaded, prior to reading data from the EEPROM. This value is truncated to the least-significant eight bits and can be set to maintain an I2C module clock of 7 to 12MHz.
-i2cclkh value	Specify the value for the I2CCLKH register. This value is loaded and takes effect after all I2C options are loaded, prior to reading data from the EEPROM.
-i2cclkl value	Specify the value for the I2CCLKL register. This value is loaded and takes effect after all I2C options are loaded, prior to reading data from the EEPROM.

Example 4-3. HEX2000.exe Command Syntax

```
C: HEX2000 GPIO34TOG.OUT -boot -gpio8 -a
Where:
- boot  Convert all sections into bootable form.
- gpio8 Use the GPIO in 8-bit mode data format. The eCAN
        uses the same data format as the GPIO in 8bit mode.
- a     Select ASCII-Hex as the output format.
```

4.9 Software

4.9.1 BOOT Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/boot

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

Chapter 5

Dual Code Security Module (DCSM)

This chapter explains the dual code security module.

5.1 Introduction

The dual code security module (DCSM) is a security feature incorporated in this device. The DCSM prevents access and visibility to on-chip secure memories (and other secure resources) by unauthorized persons. The DCSM also prevents duplication and reverse-engineering of proprietary code. The term “secure” means that access to on-chip secure memories and resources is blocked. The term “unsecure” means that access is allowed; that is, the contents of the memory can be read by any means (for example, through a debugging tool such as Code Composer Studio™ IDE.

The CSM has dual-zone security: Zone1 (Z1) and Zone2 (Z2).

5.1.1 DCSM Related Collateral

Foundational Materials

- Dual code security module (DCSM) advanced features (Video)
- Link-Pointer and Zone-Select block (Video)
- Locking and unlocking a device (Video)

Getting Started Materials

- *C2000 DCSM Security Tool Application Report*
- *C2000 Unique Device Number Application Report*
- Dual code security module (DCSM) overview (Video)
- *Secure BOOT On C2000 Device Application Report*
- [FAQ] C2000WARE: Why is my DCSM configuration causing issues in my application?

5.2 Functional Description

The security module restricts the CPU access to on-chip secure memory and resources without interrupting or stalling CPU execution. When a read occurs to a secure memory location, the read returns a zero value and CPU execution continues with the next instruction. This, in effect, blocks read and write access to secure memories through the JTAG port.

The code security mechanism offers protection for two zones, Zone1 (Z1) and Zone2 (Z2). The security mechanism for both the zones is identical. Each zone has a dedicated secure resource and allocated secure resource. The following are different secure resources available on this device:

- **OTP:** Each zone has a dedicated secure OTP (USER OTP). This contains the security configurations for the individual zone. If a zone is secure, the USER OTP content (including CSM passwords) can be read (execution not allowed) only if the zone is unlocked using the password match flow (PMF).
- **RAM:** All LSx RAMs can be secure RAM on this device. These RAMs can be allocated to either zone by configuring the respective GRABRAM locations in the USER OTP.
- **Flash Sectors:** Flash sectors of each CPU subsystems can be made secure on this device. Each Flash sector can be allocated to either zone by configuring the respective GRABSECT locations in the bank's USER OTP.
- **Secure ROM:** This device also has secure ROM on each CPU subsystem which is EXEONLY-protected. These ROM contains specific function for the user, provided by TI.

Table 5-1 shows the status of a RAM block/Flash sector based on the configuration in the GRABRAM/ GRABSECT register.

The security of each zone is enabled by a 128-bit (four 32-bit words) password (CSM password). The password for each zone is stored in USER OTP. A zone can be unsecured by executing the password match flow (PMF), described in Section 5.7.4.

There are three types of accesses:

- **Data/program reads:** Data reads to secure memory are always blocked unless the program is executing from a memory that belongs to the same zone. Data reads to unsecure memory are always allowed. Table 5-2 shows the levels of security.

- **JTAG access:** JTAG accesses are always blocked when a memory is secure.
- **Instruction fetches (calls, jumps, code executions, ISRs):** Instruction fetches are never blocked.

> **CAUTION**
> Never program any other values in these fields. Failing any of these conditions for a RAM block/ Flash sector makes that RAM block/Flash sector inaccessible.

Table 5-1. RAM/Flash Status

Zone 1 GRABRAMx/GRABSECTx Bits	Zone 2 GRABRAMx/GRABSECT Bits	Ownership and Accessibility
01	10	RAM block/Flash Sector belongs to Zone1
01	11(2)	RAM block/Flash Sector belongs to Zone1
10	01	RAM block/Flash Sector belongs to Zone2
11(1)	01	RAM block/Flash Sector belongs to Zone2
10	10	RAM block/Flash Sector is unsecure
11	11	RAM block/Flash Sector inaccessible if either of the zones is secure (CSM passwords are programmed). User must never leave these values default (11), if CSM passwords are programmed for even one zone.

(1) Zone1 must be unsecure. Assumption in this case is that user is not using Zone1 so none of the fields, including passwords, in Zone1 USER OTP are programmed by user; hence, Zone1 is always unsecure.
(2) Zone2 must be unsecure. Assumption in this case is that user is not using Zone2 so none of the fields, including passwords, in Zone2 USER OTP are programmed by user; hence, Zone2 is always unsecure.

Table 5-2. Security Levels

PMF Executed With Correct Password?	Operating Mode of the Zone	Program Fetch Location	Security Description
No	Secure	Outside secure memory	Only instruction fetches by the CPU are allowed to secure memory. In other words, code can still be executed, but not read.
No	Secure	Inside secure memory	CPU has full access (except for EXEONLY memories where read is not allowed). JTAG port cannot read the secured memory contents.
Yes	Unsecure	Anywhere	Full access for CPU and JTAG port to secure memory of that zone.

5.2.1 CSM Passwords

Unlike earlier C2000™ devices, on this device ALL_1 value (0x FFFF FFFF, 0x FFFF FFFF, 0x FFFF FFFF, 0x FFFF FFFF) for CSM password for a zone does not unsecure the zone. Instead, if for any zone the CSM password values get loaded as ALL_1 from USER OTP, the device is in BLOCKED state. Due to this reason, TI programs a few bits in the second 32-bit password value (ZxOTP_CSMPSWD1) in every zone select block of each zone with value 0. The default value for this password location is chosen in a manner that the respective ECC value remains ALL_1. Due to this, the CSMPSWD1 value programmed by TI for every zone select block is diFFerent. See Table 5-3 for ZxOTP_CSMPSWD1 value, programmed by TI on every device. Since ECC is not programmed, the user is able to change this value by flipping the bits that are 1 to 0 but leaving the bits that are already programmed by TI as 0. BOOTROM code writes the default password value into the KEYx register to unlock the device as part of device initialization sequence.

If the password locations of a zone have all 128 bits as zeros (ALL_0), that zone becomes permanently secure (LOCKED state), regardless of the contents of the CSMKEYx registers, which means the zone cannot be unlocked using PMF, see the password match flow described in Section 5.7.2. Therefore, the user can never use

ALL_0 as password. A password of ALL_0 prevents debug of secure code or reprogramming the Flash sectors. CSMKEYx registers are user-accessible registers that are used to unsecure the zones.

Table 5-3. Default Value of ZxOTP (Programmed by TI)

Zone Select Block	Zone1 USER OTP		Zone2 USER OTP	
	Address	Value	Address	Value
JLM_ENABLE (JTAGLOCK)	0x0007 8006	0xFFFF 000F	NA	NA
PSWDLOCK	0x0007 8010	0xFB7F FFFF	0x0007 8210	0x1F7F FFFF
CRCLOCK	0x0007 8012	0x7FFF FFFF	0x0007 8212	0x3FFF FFFF
JTAGPSWDH0	0x0007 8014	0x4BFF FFFF	NA	NA
JTAGPSWDH1	0x0007 8016	0x3FFF FFFF	NA	NA
Zone_Select_Block0	0x0007 8022 (CSMPSWD1)	0x4D7F FFFF	0x0007 8222 (CSMPSWD1)	0x1F7F FFFF
Zone_Select_Block0	0x0007 803E (JTAGPSWDL1)	0x2BFF FFFF	NA	NA
Zone_Select_Block1	0x0007 8042 (CSMPSWD1)	0x5F7F FFFF	0x0007 8242 (CSMPSWD1)	0xE57F FFFF
Zone_Select_Block1	0x0007 805E (JTAGPSWDL1)	0x27FF FFFF	NA	NA
Zone_Select_Block2	0x0007 8062 (CSMPSWD1)	0x1DFF FFFF	0x0007 8262 (CSMPSWD1)	0x4FFF FFFF
Zone_Select_Block2	0x0007 807E (JTAGPSWDL1)	0x7B7F FFFF	NA	NA
Zone_Select_Block3	0x0007 8082 (CSMPSWD1)	0xAF7F FFFF	0x0007 8282 (CSMPSWD1)	0xE37F FFFF
Zone_Select_Block3	0x0007 809E (JTAGPSWDL1)	0xC9FF FFFF	NA	NA
Zone_Select_Block4	0x0007 80A2 (CSMPSWD1)	0x1BFF FFFF	0x0007 82A2 (CSMPSWD1)	0x57FF FFFF
Zone_Select_Block4	0x0007 80BE (JTAGPSWDL1)	0x7D7F FFFF	NA	NA
Zone_Select_Block5	0x0007 80C2 (CSMPSWD1)	0x17FF FFFF	0x0007 82C2 (CSMPSWD1)	0x5BFF FFFF
Zone_Select_Block5	0x0007 80DE (JTAGPSWDL1)	0x6F7F FFFF	NA	NA
Zone_Select_Block6	0x0007 80E2 (CSMPSWD1)	0xBD7F FFFF	0x0007 82E2 (CSMPSWD1)	0xF17F FFFF
Zone_Select_Block6	0x0007 80FE (JTAGPSWDL1)	0x33FF FFFF	NA	NA
Zone_Select_Block7	0x0007 8102 (CSMPSWD1)	0x9F7F FFFF	0x0007 8302 (CSMPSWD1)	0x3B7F FFFF
Zone_Select_Block7	0x0007 811E (JTAGPSWDL1)	0x0FFF FFFF	NA	NA
Zone_Select_Block8	0x0007 8122 (CSMPSWD1)	0x2BFF FFFF	0x0007 8322 (CSMPSWD1)	0x8FFF FFFF
Zone_Select_Block8	0x0007 813E (JTAGPSWDL1)	0xBB7F FFFF	NA	NA
Zone_Select_Block9	0x0007 8142 (CSMPSWD1)	0x27FF FFFF	0x0007 8342 (CSMPSWD1)	0x6BFF FFFF
Zone_Select_Block9	0x0007 815E (JTAGPSWDL1)	0x5F7F FFFF	NA	NA
Zone_Select_Block10	0x0007 8162 (CSMPSWD1)	0x7B7F FFFF	0x0007 8362 (CSMPSWD1)	0x377F FFFF
Zone_Select_Block10	0x0007 817E (JTAGPSWDL1)	0x1DFF FFFF	NA	NA
Zone_Select_Block11	0x0007 8182 (CSMPSWD1)	0xC9FF FFFF	0x0007 8382 (CSMPSWD1)	0x9BFF FFFF
Zone_Select_Block11	0x0007 819E (JTAGPSWDL1)	0xAF7F FFFF	NA	NA
Zone_Select_Block12	0x0007 81A2 (CSMPSWD1)	0x7D7F FFFF	0x0007 83A2 (CSMPSWD1)	0x2F7F FFFF
Zone_Select_Block12	0x0007 81BE (JTAGPSWDL1)	0x1BFF FFFF	NA	NA
Zone_Select_Block13	0x0007 81C2 (CSMPSWD1)	0x6F7F FFFF	0x0007 83C2 (CSMPSWD1)	0xCB7F FFFF
Zone_Select_Block13	0x0007 81DE (JTAGPSWDL1)	0x17FF FFFF	NA	NA
Zone_Select_Block14	0x0007 81E2 (CSMPSWD1)	0x33FF FFFF	0x0007 83E2 (CSMPSWD1)	0x97FF FFFF
Zone_Select_Block14	0x0007 81FE (JTAGPSWDL1)	0xBD7F FFFF		

5.2.2 Emulation Code Security Logic (ECSL)

In addition to the CSM, the emulation code security logic (ECSL) has been implemented using a 64-bit password (part of existing CSM password) for each zone to prevent unauthorized users from stepping through secure code. A halt in secure code while the emulator is connected trips the ECSL and break the emulation connection. To allow emulation of secure code, while maintaining the CSM protection against secure memory reads, you must write the correct 64-bit password into the CSMKEY (0/1) registers, which matches the password value stored in the USER OTP of that zone. This disables the ECSL for the specific zone.

When initially debugging a device with the password locations in OTP programmed (secured), the emulator takes some time to take control of the CPU. During this time, the CPU starts running and can execute an instruction that performs an access to a protected ECSL area and if the CPU is halted when the program counter (PC) is pointing to a secure location, the ECSL trips and causes the emulator connection to be broken.

One answer to this problem is to use the Wait Boot Mode boot option. In this mode, the CPU is in a loop and does not jump to the user application code. Using this BOOTMODE, you can connect to CCS and debug the code.

5.2.3 CPU Secure Logic

The CPU Secure Logic (CPUSL) on this device prevents a hacker from reading the CPU registers in a watch window while code is running in a secure zone. All accesses to CPU registers when the PC points to a secure location are blocked by this logic. The only exception to this is read access to the PC. It is highly recommended not to write into the CPU register in this case, because proper code execution can get affected. If the CSM is unlocked using the CSM password match flow, the CPUSL logic also gets disabled.

5.2.4 Execute-Only Protection

To achieve a higher level of security on secure Flash sectors and RAM blocks that store critical user code (instruction opcodes), the Execute-Only protection feature is provided. When the Execute-Only protection is turned on for any secure Flash sector or RAM block, data reads to that Flash sector or RAM block are disallowed from any code (even from secure code). Execute-only protection for a Flash sector and RAM block can be turned on by configuring the bit field associated for that particular sector/RAM block in the zone's (which has ownership of that sector/RAM block) EXEONLYSECT and EXEONLYRAM register, respectively.

5.2.5 Password Lock

The password locations in USER OTP for each zone can be locked by programming the zone’s PSWDLOCK field with any value other than 1111 (0xF) at the PSWDLOCK location in OTP. Until the passwords of a zone are locked, password locations are not secure and can be read from the debugger as well as code running from non-secure memory. This feature can be used by the user to avoid accidental locking of the zone while programming the Flash sectors during the software development phase. On a new device, the value for password lock fields for all zones at the PSWDLOCK location in OTP is 1111, which means the password for all zones is unlocked.

Note

Password unlock only makes password locations non-secure. All other secure memories remains secure as per security settings. Since password locations are non-secure, anyone can read the password and make the zone unsecure by running through PMF, the user must program the PSWDLOCK locations to lock the password before sending the device in the field.

5.2.6 JTAGLOCK

To disable the JTAG access on a device to avoid any debug access, use the JTAGLOCK feature on this device. Follow a two step process to enable the JTAGLOCK feature (both steps can be performed at the same time):

1. Program the JTAG passwords. This device has a 128-bit JTAG password that needs to be programmed in Z1 USER OTP. JTAG passwords are split into two parts, JTAGPSWDH and JTAGPSWDL. JTAGPSWDH is part of Z1 USER OTP header and JTAGPSWDL is part of Z1 Zone Select Block (ZSB). What this means is program JTAGPSWDH once and change the JTAGPSWDL multiple times, if needed. The Code Composer Studio™ IDE has an integrated tool that you use to unlock the JTAGLOCK on the device.
2. After programming the JTAG passwords, enable the JTAGLOCK module (JLM) by programming bit [3:0] of Z1OTP_JLM_ENABLE with any value other than 0xF. It is recommended to program all four bits with a value 0x0.

5.2.7 Link Pointer and Zone Select

For each of the two security zones, a dedicated OTP block exists that holds the configuration related to zone's security. The following are user programmable configurations:

- ZxOTP_LINKPOINTER1
- ZxOTP_LINKPOINTER2
- ZxOTP_LINKPOINTER3
- Z1OTP_JLM_ENABLE
- ZxOTP_GPREG1
- ZxOTP_GPREG2
- ZxOTP_GPREG3
- ZxOTP_GPREG4
- ZxOTP_PSWDLOCK
- ZxOTP_CRCLOCK
- Z1OTP_JTAGPSWDH
- Z1OTP_CMACKEY
- ZxOTP_CSMPSWD0
- ZxOTP_CSMPSWD1
- ZxOTP_CSMPSWD2
- ZxOTP_CSMPSWD3
- ZxOTP_GRABSECT1
- ZxOTP_GRABSECT2
- ZxOTP_GRABSECT3
- ZxOTP_GRABRAM1
- ZxOTP_EXEONLYSECT1
- ZxOTP_EXEONLYSECT2
- ZxOTP_EXEONLYRAM1
- Z1OTP_JTAGPSWDL

Since OTP cannot be erased, the following configurations are placed in zone select blocks of each zone's OTP Flash of both the banks:

- ZxOTP_CSMPSWD0
- ZxOTP_CSMPSWD1
- ZxOTP_CSMPSWD2
- ZxOTP_CSMPSWD3
- ZxOTP_GRABSECT1
- ZxOTP_GRABSECT2
- ZxOTP_GRABSECT3
- ZxOTP_GRABRAM1
- ZxOTP_EXEONLYSECT1
- ZxOTP_EXEONLYSECT2
- ZxOTP_EXEONLYRAM1
- Z1OTP_JTAGPSWDL

The location of the valid zone select block in OTP is decided based on the value of three 14-bit link pointers (Zx-LINKPOINTERx) programmed in the OTP of each zone. All OTP locations except link pointers and Z1OTP_JLM_ENABLE locations are protected with ECC. Since the link pointer locations are not protected with ECC, three link pointers are provided that need to be programmed with the same value. The final value of the link pointer is resolved in hardware, when a dummy read is done to all the link pointers, by comparing all the three values (bit-wise voting logic). Since in OTP, a 1 can be flipped by the user to 0, but a 0 cannot be flipped to a 1 (no erase operation for OTP), the most-significant bit position in the resolved link pointer that is 0 defines the valid base address for the zone select block. While generating the final link pointer value, if the bit pattern is not one of those listed in Figure 5-1, the final link pointer value becomes All_1 (0xFFFF_FFFF), which selects the Zone-Select-Block1 (also known as the default zone select block).

Zx-LINKPOINTER	Selected ZSB	Zone1 ZSB Address	Zone2 ZSB Address
32xxxxxxxxxxxxxxxxxx11111111111111	ZSB1	0x78020	0x78220
32xxxxxxxxxxxxxxxxxx11111111111110	ZSB2	0x78040	0x78240
32xxxxxxxxxxxxxxxxxx11111111111100	ZSB3	0x78060	0x78260
32xxxxxxxxxxxxxxxxxx11111111111000	ZSB4	0x78080	0x78280
32xxxxxxxxxxxxxxxxxx11111111110000	ZSB5	0x780a0	0x782a0
32xxxxxxxxxxxxxxxxxx11111111100000	ZSB6	0x780c0	0x782c0
32xxxxxxxxxxxxxxxxxx11111111000000	ZSB7	0x780e0	0x782e0
32xxxxxxxxxxxxxxxxxx11111110000000	ZSB8	0x78100	0x78300
32xxxxxxxxxxxxxxxxxx11111100000000	ZSB9	0x78120	0x78320
32xxxxxxxxxxxxxxxxxx11111000000000	ZSB10	0x78140	0x78340
32xxxxxxxxxxxxxxxxxx11110000000000	ZSB11	0x78160	0x78360
32xxxxxxxxxxxxxxxxxx11100000000000	ZSB12	0x78180	0x78380
32xxxxxxxxxxxxxxxxxx11000000000000	ZSB13	0x781a0	0x783a0
32xxxxxxxxxxxxxxxxxx10000000000000	ZSB14	0x781c0	0x783c0
32xxxxxxxxxxxxxxxxxx00000000000000	ZSB15	0x781e0	0x783e0

Figure 5-1. Storage of Zone-Select Bits in OTP

Note

Address locations for other security settings that are not part of Zone Select blocks can be programmed only once; therefore, you must program the address locations toward the end of the development cycle.

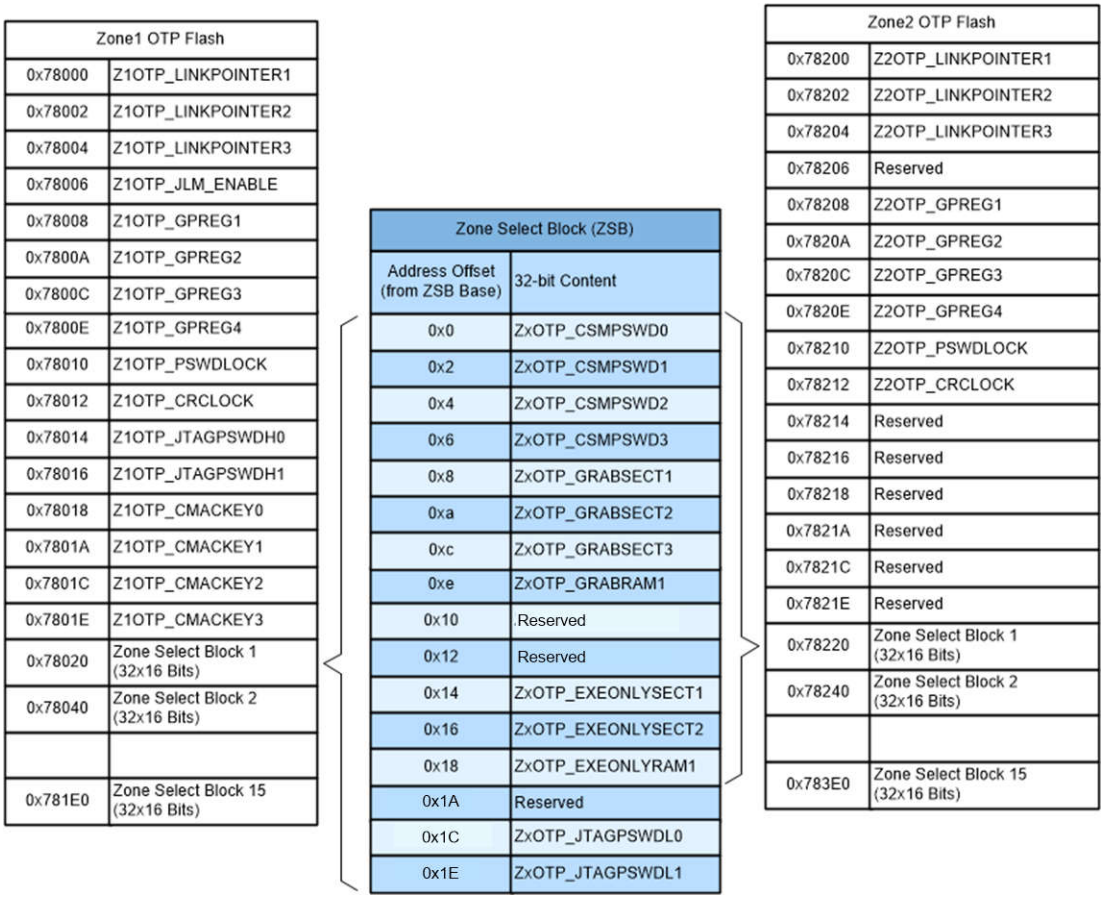


Figure 5-2. Location of Zone-Select Block Based on Link-Pointer

CAUTION

USER OTP is ECC protected. You must program the ECC value while programming the security setting in USER OTP. Failing to program the correct ECC value causes the device to be blocked permanently and the device needs to be replaced.

5.2.8 C Code Example to Get Zone Select Block Addr for Zone1

```
unsigned long LinkPointer;
unsigned long *Zone1SelBlockPtr;
int Bitpos = 13;
int ZeroFound = 0;
// Read Z1-Linkpointer register of DCSM module.
LinkPointer = *(unsigned long *)0x5F000;
// Bits 31 to 15 as most-sigificant 0 are reserved LinkPointer options
LinkPointer = LinkPointer << 18;
while ((ZeroFound == 0) && (bitpos > -1))
{
    if ((LinkPointer & 0x80000000) == 0)
    {
        ZeroFound = 1;
        Zone1SelBlockPtr = (unsigned long *)(0x78000 + ((bitpos + 2)*32));
    }
    Else
    {
        bitpos--;
        LinkPointer = LinkPointer << 1;
    }
}
if (ZeroFound == 0)
{
    //Default in case there is no zero found.
    Zone1SelBlockPtr = (unsigned long *)0x78020;
}
```

5.3 Flash and OTP Erase/Program

On this device, OTP as well as normal Flash, are secure resources. Each zone has a dedicated OTP, whereas normal Flash sectors can be allocated to any zone based on the value programmed in the GRABSECTx location in OTP. Each zone has 128-bit CSM passwords. Read and write accesses are not allowed to resources assigned to Z1 by code running from memory allocated to Z2 and conversely. Before programming any secure Flash sector, the user must either unlock the zone to which that particular sector belongs using PMF or execute the Flash programming code from secure memory that belongs to the same zone. The same is the case for erasing any secure Flash sector. To program the security settings in OTP Flash, the user must unlock the CSM of the respective zone. Unless the zone is unlocked, security settings in OTP Flash can not be updated. The OTP content cannot be erased.

A semaphore mechanism is provided to avoid the program/erase conflict between Z1 and Z2. A zone needs to grab this semaphore to successfully complete the erase/program operation on the secure Flash sectors allocated to that zone. A semaphore can be grabbed by a zone by writing the appropriate value in the SEM field of the FLSEM register. For further details of this field, see the register description.

5.4 Secure Copy Code

In some applications, the user wants to copy the code from secure Flash to secure RAM for better performance. The user cannot do this for EXEONLY Flash sectors because EXEONLY secure memories cannot be read from anywhere. TI provides specific “Secure Copy Code” library functions for ZONE1 to enable the user to copy content from EXEONLY secure Flash sectors to EXEONLY RAM blocks. These functions do the copy-code operation in a highly secure environment and allow a copy to be performed only when the following conditions are met:

- The secure RAM block and the secure Flash sector belong to the same zone.
- Both the secure RAM block and the secure Flash sector have EXEONLY protection enabled.

For further usage of these library functions, see the device-specific Boot ROM documentation.

5.5 SecureCRC

Since reads from EXEONLY memories are not allowed, the user cannot calculate the CRC on content in EXEONLY memories using the CRC engine available on this device (for example, VCUCRC, GCRC) or software. In some safety-critical applications, the user has to calculate the CRC even on these memories. To enable this without compromising on security, TI provides specific “SecureCRC” library functions for each zone. These functions do the CRC calculation in highly secure environment and allow a CRC calculation to be performed only when the following conditions are met:

- The source address is modulo the number of words (based on length_id) for which the CRC needs to be calculated.
- The destination address belongs to the same zone as the source address.

For further usage of these library functions, see the device-specific Boot ROM documentation.

Note

The user must disable all the interrupts before calling the secure functions in ROM. If there is a vector fetch during secure function execution, the CPU gets reset immediately.

Disclaimer: The Code Security Module (CSM) included on this device was designed to password protect the data stored in the associated memory and is warranted by Texas Instruments (TI), in accordance with its standard terms and conditions, to conform to TI's published specifications for the warranty period applicable for this device. TI DOES NOT, HOWEVER, WARRANT OR REPRESENT THAT THE CSM CANNOT BE COMPROMISED OR BREACHED OR THAT THE DATA STORED IN THE ASSOCIATED MEMORY CANNOT BE ACCESSED THROUGH OTHER MEANS. MOREOVER, EXCEPT AS SET FORTH ABOVE, TI MAKES NO WARRANTIES OR REPRESENTATIONS CONCERNING THE CSM OR OPERATION OF THIS DEVICE, INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL TI BE LIABLE FOR ANY CONSEQUENTIAL, SPECIAL, INDIRECT, INCIDENTAL, OR PUNITIVE DAMAGES, HOWEVER CAUSED, ARISING IN ANY WAY OUT OF YOUR USE OF THE CSM OR THIS DEVICE, WHETHER OR NOT TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCLUDED DAMAGES INCLUDE, BUT ARE NOT LIMITED TO LOSS OF DATA, LOSS OF GOODWILL, LOSS OF USE OR INTERRUPTION OF BUSINESS OR OTHER ECONOMIC LOSS.

5.6 CSM Impact on Other On-Chip Resources

CAUTION

The order of initialization matters; hence, if a memory watch window with the USER OTP address is opened in the debugger (CCS IDE), the security initialization can occur in an incorrect order locking the device down. To avoid this, do not keep a memory window with USER OTP address opened in the debugger (CCS IDE) when performing a reset.

Note

Security initialization is done by BOOTROM code on all the resets (as part of device initialization) that assert SYSRSn. This is not part of the user application code.

On this device, some of the memories are not secure. To avoid any potential hacking when the device is in the default state (post reset), accesses (all types) to all memories (secure as well as non-secure, except BOOT-ROM and OTP) are disabled until proper security initialization is done. This means that after reset none of the memory resources except BOOT_ROM and OTP is accessible to the user.

The following steps are required by CPU after reset (any type of reset) to initialize the security on device.

Security Initialization

- Dummy Read to address location of SECDC (0x703F0, TI-reserved register) in TI OTP
- Dummy Read to address location of Z1OTP_LINKPOINTER1 in Z1 OTP
- Dummy Read to address location of Z1OTP_LINKPOINTER2 in Z1 OTP
- Dummy Read to address location of Z1OTP_LINKPOINTER3 in Z1 OTP
- Dummy Read to address location of Z2OTP_LINKPOINTER1 in Z2 OTP
- Dummy Read to address location of Z2OTP_LINKPOINTER2 in Z2 OTP
- Dummy Read to address location of Z2OTP_LINKPOINTER3 in Z2 OTP
- Dummy Read to address location Z1OTP_JLM_ENABLE in Z1 OTP
- Dummy Read to address location of Z1OTP_GPREG1, Z1OTP_GPREG2, Z1OTP_GPREG3, Z1OTP_GPREG4 in Z1 OTP
- Dummy Read to address location of Z1OTP_PSWDLOCK in Z1 OTP
- Dummy Read to address location of Z1OTP_CRCLOCK in Z1 OTP
- Dummy Read to address location of Z1OTP_JTAGPSWDH0, Z1OTP_JTAGPSWDH1 in Z1 OTP
- Dummy Read to address location of Z2OTP_GPREG1, Z2OTP_GPREG2, Z2OTP_GPREG3, Z2OTP_GPREG4 in Z2 OTP
- Dummy Read to address location of Z2OTP_PSWDLOCK in Z2 OTP
- Dummy Read to address location of Z2OTP_CRCLOCK in Z2 OTP
- Read to memory map register of Z1_LINKPOINTER in DCSM module to calculate the address of zone select block for Z1
- Dummy read to address location of Z1OTP_GRABSECT1, Z1OTP_GRABSECT2, Z1OTP_GRABSECT3 in Z1 OTP
- Dummy read to address location of Z1OTP_GRABRAM1
- Dummy read to address location of Z1OTP_EXEONLYSECT1, Z1OTP_EXEONLYSECT2 in Z1 OTP
- Dummy read to address location of Z1OTP_EXEONLYRAM1 in Z1 OTP
- Dummy Read to address location of Z1OTP_JTAGPSWDL0, Z1OTP_JTAGPSWDL1 in Z1 OTP
- Read to memory map register of Z2_LINKPOINTER in DCSM module to calculate the address of zone select block for Z2
- Dummy read to address location of Z2OTP_GRABSECT1, Z2OTP_GRABSECT2, Z2OTP_GRABSECT3 in Z2 OTP
- Dummy read to address location of Z2OTP_GRABRAM1
- Dummy read to address location of Z2OTP_EXEONLYSECT1, Z2OTP_EXEONLYSECT2 in Z2 OTP
- Dummy read to address location of Z2OTP_EXEONLYRAM1 in Z2 OTP

5.6.1 RAMOPEN

RAMOPEN feature on this device enables a user to unsecure all of the secure RAM blocks without unlocking the zone security. Afterward, the RAMs can be converted back into secure RAMs without issuing reset to device. However, the previous contents of the secure RAM are not maintained.

This is a useful feature for Flash programming tools (for example, CCS Flash plugin, serial Flash programmer, and so on), which need to download Flash APIs and data on RAMs to carry out the Flash program/erase operations to unsecure Flash regions. Since debugger and boot loader writes are considered unsecure, the writes require either unsecure RAM or unlocked secure RAM to download the APIs and data to. However, the user can not always have the password required to unlock one or both of the security zones. This can be particularly problematic in a device with limited RAM where the majority of the RAM used during application runtime is designated as secure RAM.

The RAMOPEN feature allows the Flash programming tools to temporarily convert the secured RAM blocks into unsecured RAM blocks to load the API/Data and after Flash program/erase operation is completed, turn them back into secured RAM blocks.

The feature works as:

- RAMOPEN feature is enabled by setting RAMOPENFRC.SET bit to 1
- This clears the content of all the secure RAM (RAMINIT is performed in hardware).
- After completion of RAMINIT operation, RAMOPENSTAT.RAMOPEN bit get set to 1 (RAMOPEN feature is enabled and all secure RAM blocks are unsecure).
- To disable RAMOPEN feature, set RAMOPENCLR.CLEAR bit to 1.
- This again clears the content of all secure RAM (RAMINIT is performed in hardware).
- After completion of RAMINIT operation, RAMOPENSTAT.RAMOPEN bit gets cleared to 0 (RAMOPEN feature is disabled and all secure RAM blocks are again secure).

5.7 Incorporating Code Security in User Applications

Code security is typically not required in the development phase of a project. However, security is needed once a robust code is developed for a zone. Before such a code is programmed in the Flash memory, a CSM password must be chosen to secure the zone. Once a CSM password is in place for a zone, the zone is secured (programming a password at the appropriate locations and either performing a device reset or setting the FORCESEC bit (Zx_CR.31) is the action that secures the device). From that time on, access to debug the contents of secure memory by any means (using JTAG, code running off external/on-chip memory, and so forth) requires a valid password. A password is not needed to run the code out of secure memory (such as in a typical end-user usage); however, access to secure memory contents for debug purposes requires a password.

5.7.1 Environments That Require Security Unlocking

The following are the typical situations under which unsecuring the zone can be required:

- Code development using debuggers (such as Code Composer Studio™ IDE). This is the most common environment during the design phase of a product.
- Flash programming using TI's Flash utilities such as Code Composer Studio IDE On-Chip Flash Programmer plug-in or the UniFlash tool. Flash programming is common during code development and testing. Once the user supplies the necessary password, the Flash utilities disable the security logic before attempting to program the Flash. In custom programming that uses the Flash API supplied by TI, unlocking the CSM can be avoided by executing the Flash programming algorithms from secure memory.
- Custom environment defined by the application. In addition to the above, access to secure memory contents can be required in situations such as:
  - Using the on-chip bootloader to load code or data into secure SRAM or to erase and program the Flash.
  - Executing code from on-chip unsecure memory and requiring access to secure memory for the lookup table. This is not a suggested operating condition as supplying the password from external code can compromise code security.

The unsecuring sequence is identical in all the above situations. This sequence is referred to as the password match flow (PMF) for simplicity. Figure 5-3 explains the sequence of operation that is required every time the user attempts to unsecure a particular zone. A code example is listed for clarity.

5.7.2 CSM Password Match Flow

Password match flow (PMF) is essentially a sequence of four dummy reads from password locations (PWL) followed by four writes (32-bit writes) to CSMKEY(0/1/2/3) registers. Figure 5-3 shows how PMF helps to initialize the security logic registers and disable security logic.

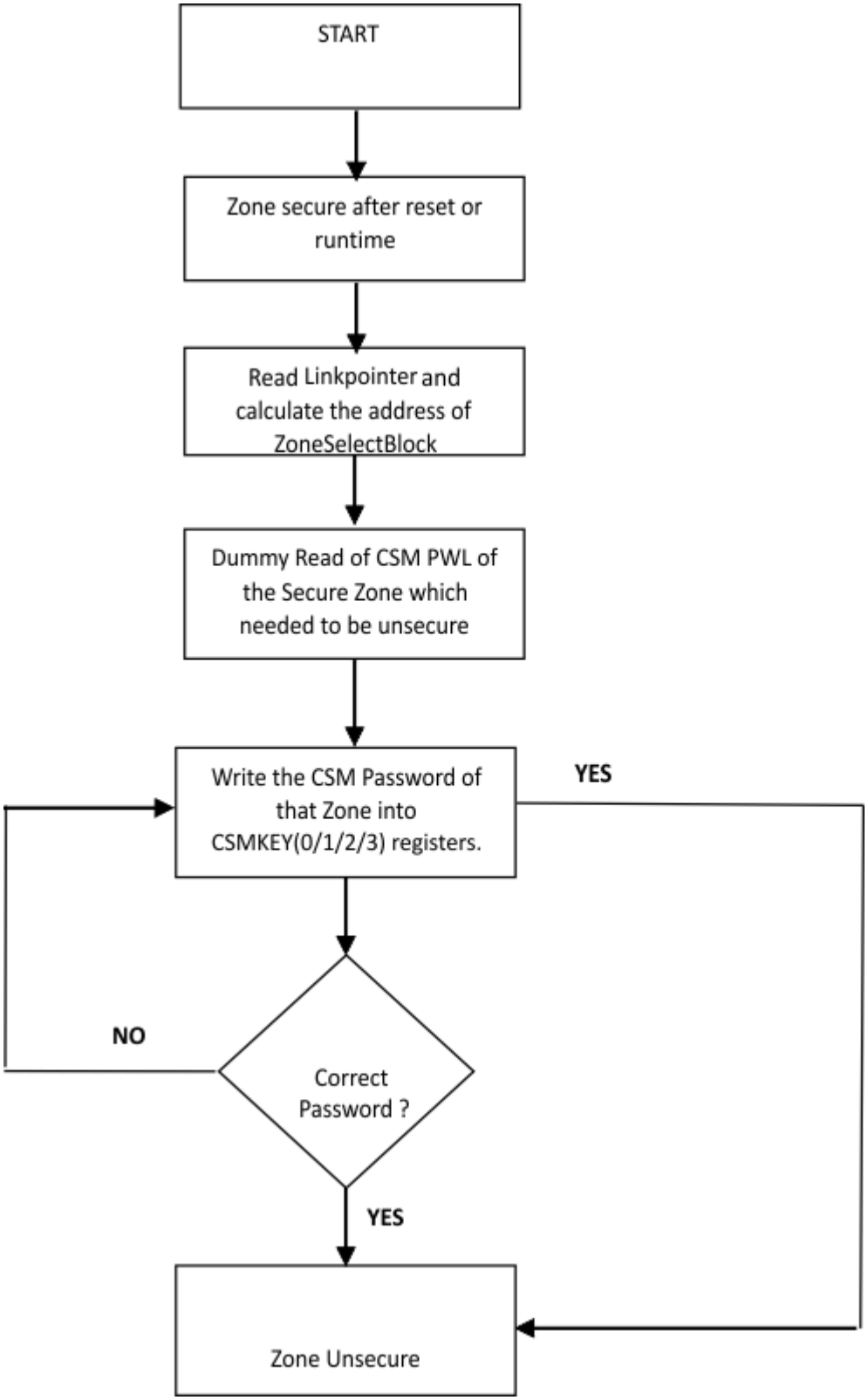


Figure 5-3. CSM Password Match Flow (PMF)

5.7.3 C Code Example to Unsecure C28x Zone1

```
volatile long int *CSM = (volatile long int *)5F090; //CSM register file volatile
long int *CSMPWL = (volatile long int *)0x78020; //CSM Password location (assuming default Zone
select block)
volatile int tmp;
int I;
// Read the 128-bits of the CSM password locations (PWL)
//
for (I=0;I<4; I++) tmp = *CSMPWL++;
// Write the 128-bit password to the CSMKEY registers
// If this password matches that stored in the
// CSLPWL, then the CSM becomes unsecure. If the password does not
// match, then the zone remains secure.
// An example password of: // 0x11112222333344445555666677778888 is used.
*CSM++ = 0x22221111; // Register Z1_CSMKEY0 at 0x5F090
*CSM++ = 0x44443333; // Register Z1_CSMKEY1 at 0x5F092
*CSM++ = 0x66665555; // Register Z1_CSMKEY2 at 0x5F094
*CSM++ = 0x88887777; // Register Z1_CSMKEY3 at 0x5F096
```

5.7.4 C Code Example to Resecure C28x Zone1

```
volatile int *Z1_CR = 0x5F019; //CSMSCR register
//Set FORCESEC bit
*Z1_CR = 0x8000;
```

5.7.5 Environments That Require ECSL Unlocking

The following are the typical situations under which unsecuring can be required:

- The user develops some main IP, and then outsources peripheral functions to a subcontractor who must be able to run the user code during debug and can halt while the main IP code is running. If ECSL is not unlocked, then Code Composer Studio™ IDE connections get disconnected, which can be inconvenient for the user. Note that unlocking ECSL does not enable access to the secure code but only avoids disconnection of Code Composer Studio™ IDE (JTAG).

5.7.6 ECSL Password Match Flow

A password match flow (PMF) is essentially a sequence of eight dummy reads from password locations (PWL) followed by two writes to KEY registers. Figure 5-4 shows how the PMF helps to initialize the security logic registers and disable security logic.

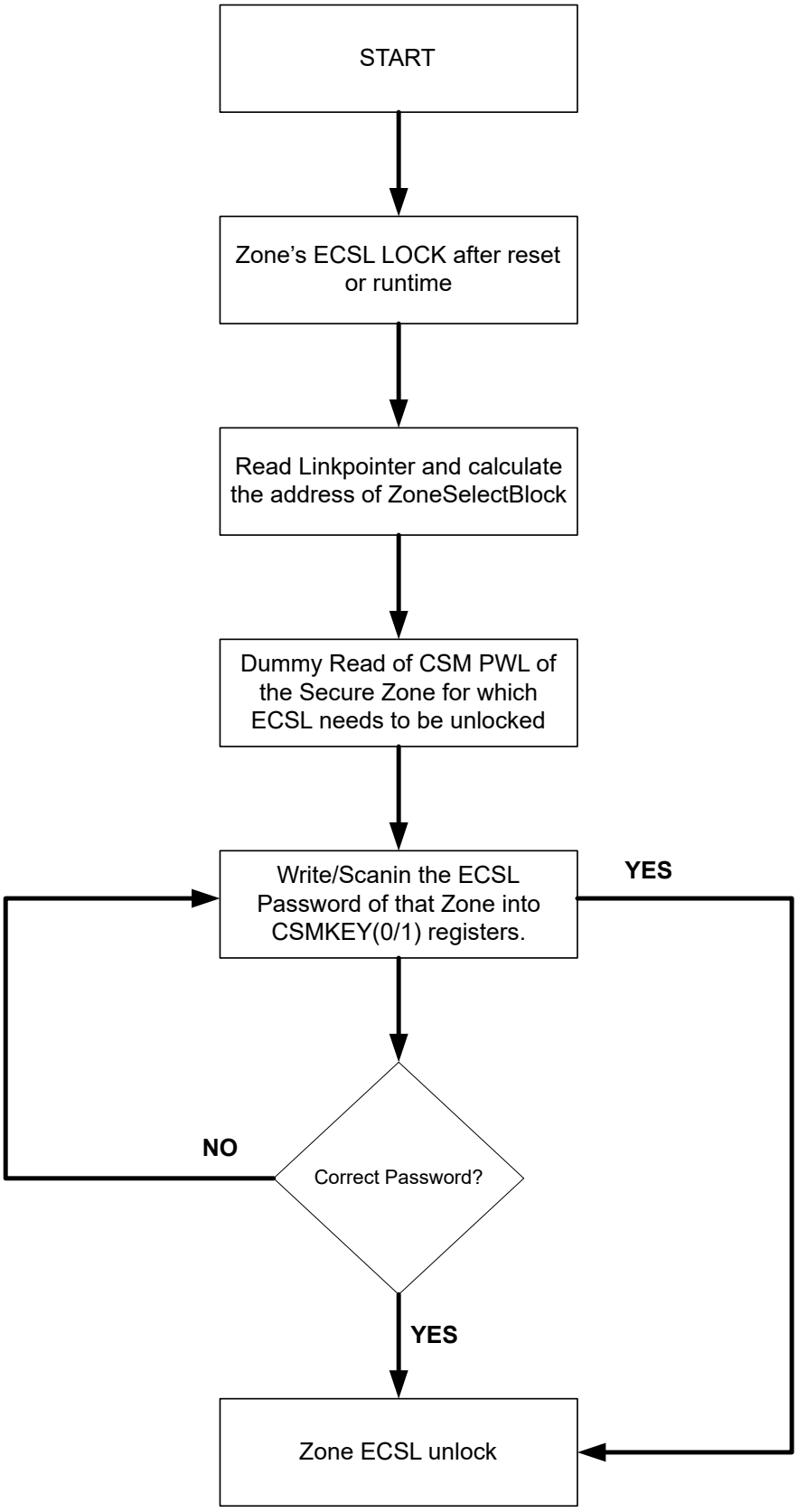


Figure 5-4. ECSL Password Match Flow (PMF)

5.7.7 ECSL Disable Considerations for any Zone

A zone with ECSL enabled must have a predetermined ECSL password stored in the ECSL password locations in Flash (same as lower 64-bits of CSM passwords). Perform the following steps to disable the ECSL for any particular zone:

- Perform a dummy read of CSM password locations of that Zone.
- Write the password into the CSMKEY0/1 registers, corresponding to that Zone.
- If the password is correct, the ECSL gets disabled; otherwise, the ECSL stays enabled.

5.7.7.1 C Code Example to Disable ECSL for C28x Zone1

```
volatile long int *ECSL = (volatile int *)0x5F090; //ECSL register file
volatile long int *ECSLPWL = (volatile int *)0x78028; //ECSL Password location (assuming default
Zone sel block)
volatile int tmp;
int I;
// Read the 64-bits of the password locations (PWL).
for (I=0;I<2; I++) tmp = *ECSLPWL++;
// Write the 64-bit password to the CSMKEYx registers
// If this password matches that stored in the
// CSMPWL, then ECSL gets disabled. If the password does not
// match, then the zone remains secure.
// An example password of: // 0x1111222233334444 is used.
*ECSL++ = 0x22221111; // Register Z1_CSMKEY0 at 0x5F090
*ECSL++ = 0x44443333; // Register Z1_CSMKEY1 at 0x5F092
```

5.7.8 Device Unique ID

TI OTP contains a 256-bit value that is made up of both random and sequential parts. This value can be used as a seed for code encryption. The starting address of the value is 0x71140. The first 192 bits are random, the next 32 bits are sequential, and the last 32 bits are a checksum value.

5.8 Software

5.8.1 DCSM Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/dcsm

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

5.8.1.1 Empty DCSM Tool Example

FILE: dcsm_security_tool.c

This example is an empty project setup for DCSM Tool and Driverlib development. For guidance refer to: *C2000 DCSM Security Tool*

5.9 DCSM Registers

This Section describes the DCSM Registers.

5.9.1 DCSM Base Address Table

Table 5-4. DCSM Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
DcsmZ1Regs	DCSM_Z1_REGS	DCSM_Z1_BASE	0x0005_F000	-	YES
DcsmZ2Regs	DCSM_Z2_REGS	DCSM_Z2_BASE	0x0005_F080	-	YES
DcsmCommonRegs	DCSM_COMMON_REGS	DCSMCOMMON_BASE	0x0005_F0C0	-	YES
DcsmZ1OtpRegs	DCSM_Z1_OTP	DCSM_Z1OTP_BASE	0x0007_8000	-	-
DcsmZ2OtpRegs	DCSM_Z2_OTP	DCSM_Z2OTP_BASE	0x0007_8200	-	-

5.9.2 DCSM_Z1_REGS Registers

Table 5-5 lists the memory-mapped registers for the DCSM_Z1_REGS registers. All register offset addresses not listed in Table 5-5 should be considered as reserved locations and the register contents should not be modified.

Table 5-5. DCSM_Z1_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	Z1_LINKPOINTER	Zone 1 Link Pointer	
2h	Z1_OTPSECLOCK	Zone 1 OTP Secure Lock	
4h	Z1_JLM_ENABLE	Zone 1 JTAGLOCK Enable Register	
6h	Z1_LINKPOINTERERR	Link Pointer Error	
8h	Z1_GPREG1	Zone 1 General Purpose Register-1	
Ah	Z1_GPREG2	Zone 1 General Purpose Register-2	
Ch	Z1_GPREG3	Zone 1 General Purpose Register-3	
Eh	Z1_GPREG4	Zone 1 General Purpose Register-4	
10h	Z1_CSMKEY0	Zone 1 CSM Key 0	
12h	Z1_CSMKEY1	Zone 1 CSM Key 1	
14h	Z1_CSMKEY2	Zone 1 CSM Key 2	
16h	Z1_CSMKEY3	Zone 1 CSM Key 3	
18h	Z1_CR	Zone 1 CSM Control Register	
1Ah	Z1_GRABSECT1R	Zone 1 Grab Flash Status Register 1	
26h	Z1_EXEONLYSECT1R	Zone 1 Execute Only Flash Status Register 1	
2Eh	Z1_JTAGKEY0	JTAG Unlock Key Register 0	
30h	Z1_JTAGKEY1	JTAG Unlock Key Register 1	
32h	Z1_JTAGKEY2	JTAG Unlock Key Register 2	
34h	Z1_JTAGKEY3	JTAG Unlock Key Register 3	
36h	Z1_CMACKEY0	Secure Boot CMAC Key Status Register 0	
38h	Z1_CMACKEY1	Secure Boot CMAC Key Status Register 1	
3Ah	Z1_CMACKEY2	Secure Boot CMAC Key Status Register 2	
3Ch	Z1_CMACKEY3	Secure Boot CMAC Key Status Register 3	

Complex bit access types are encoded to fit into small table cells. Table 5-6 shows the codes that are used for access types in this section.

Table 5-6. DCSM_Z1_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

5.9.2.1 Z1_LINKPOINTER Register (Offset = 0h) [Reset = FFFFC000h]

Z1_LINKPOINTER is shown in Figure 5-5 and described in Table 5-7.

Return to the Summary Table.

Zone 1 Link Pointer

Figure 5-5. Z1_LINKPOINTER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		LINKPOINTER													
R-0003FFFFh																		R-0h													

Table 5-7. Z1_LINKPOINTER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0003FFFFh	Reserved
13-0	LINKPOINTER	R	0h	This is resolved Link-Pointer value which is generated by looking at the three physical Link-Pointer values loaded from OTP Reset type: SYSRSn

5.9.2.2 Z1_OTPSECLOCK Register (Offset = 2h) [Reset = 00000001h]

Z1_OTPSECLOCK is shown in Figure 5-6 and described in Table 5-8.

Return to the Summary Table.

Zone 1 OTP Secure Lock

Figure 5-6. Z1_OTPSECLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				RESERVED			
R-0h				R-0h			
7	6	5	4	3	2	1	0
PSWDLOCK				RESERVED			JTAGLOCK
R-0h				R-0h			R-1h

Table 5-8. Z1_OTPSECLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	RESERVED	R	0h	Reserved
11-8	RESERVED	R	0h	Reserved
7-4	PSWDLOCK	R	0h	Value in this field gets loaded from Z1_PSWDLOCK[3:0] when a read is issued to address location of Z1_PSWDLOCK in OTP. 1111 : CSM password locations in OTP are not protected and can be read from debugger as well as code running from anywhere. Other Value : CSM password locations in OTP are protected and can't be read without unlocking CSM of that zone. Reset type: XRSn
3-1	RESERVED	R	0h	Reserved
0	JTAGLOCK	R	1h	Reflects the state of the JTAGLOCK feature. 0 : JTAG is not locked 1 : JTAG is locked Reset type: PORESETn

5.9.2.3 Z1_JLM_ENABLE Register (Offset = 4h) [Reset = 0000000Fh]

Z1_JLM_ENABLE is shown in Figure 5-7 and described in Table 5-9.

Return to the Summary Table.

Zone 1 JTAGLOCK Enable Register

Figure 5-7. Z1_JLM_ENABLE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				Z1_JLM_ENABLE			
R-0-0h				R-Fh			

Table 5-9. Z1_JLM_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3-0	Z1_JLM_ENABLE	R	Fh	Zone1 JLM_ENABLE register. The value in this field gets loaded from Z1OTP_JLM_ENABLE[3:0] when a read is issued to address location of Z1OTP_JLM_ENABLE in OTP. If Z1OTP_JLM_ENABLE[31:0] is equal to all ones during the load, the JTAGLOCK is not bypassed (is enabled). If the value of Z1OTP_JLM_ENABLE[31:0] is not all ones during the load, the JTAGLOCK is governed as follows by the Z1_JLM_ENABLE bits: 1111 : JTAG/Emulation access is allowed (JTAGLOCK is not enabled) Other values: JTAGLOCK is governed by the JTAGKEY==JTAGPSWD match condition Reset type: PORESETn

5.9.2.4 Z1_LINKPOINTERERR Register (Offset = 6h) [Reset = 00000000h]

Z1_LINKPOINTERERR is shown in Figure 5-8 and described in Table 5-10.

Return to the Summary Table.

Link Pointer Error

Figure 5-8. Z1_LINKPOINTERERR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		Z1_LINKPOINTERERR													
R-0-0h		R-0h													

Table 5-10. Z1_LINKPOINTERERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	Z1_LINKPOINTERERR	R	0h	These bits indicate errors during formation of the resolved Link-Pointer value after the three physical Link-Pointer values loaded of OTP in flash. 0 : No Error. Other : Error on bit positions which is set to 1. Reset type: SYSRSn

5.9.2.5 Z1_GPREG1 Register (Offset = 8h) [Reset = 00000000h]

Z1_GPREG1 is shown in Figure 5-9 and described in Table 5-11.

Return to the Summary Table.

Zone 1 General Purpose Register-1

Figure 5-9. Z1_GPREG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPREG1																															
R-0h																															

Table 5-11. Z1_GPREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPREG1	R	0h	This field gets loaded with the contents of Z1OTP_GPREG1 locations in Zone-1-USER-OTP when a dummy read is issued to that address. Users can use this register to load any general purpose non-voliate content from USER-OTP of Zone-1 Reset type: SYSRSn

5.9.2.6 Z1_GPREG2 Register (Offset = Ah) [Reset = 00000000h]

Z1_GPREG2 is shown in Figure 5-10 and described in Table 5-12.

Return to the Summary Table.

Zone 1 General Purpose Register-2

Figure 5-10. Z1_GPREG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPREG2																															
R-0h																															

Table 5-12. Z1_GPREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPREG2	R	0h	This field gets loaded with the contents of Z1OTP_GPREG2 locations in Zone-1-USER-OTP when a dummy read is issued to that address. Users can use this register to load any general purpose non-voliate content from USER-OTP of Zone-1 Reset type: SYSRSn

5.9.2.7 Z1_GPREG3 Register (Offset = Ch) [Reset = 00000000h]

Z1_GPREG3 is shown in Figure 5-11 and described in Table 5-13.

Return to the Summary Table.

Zone 1 General Purpose Register-3

Figure 5-11. Z1_GPREG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPREG3																															
R-0h																															

Table 5-13. Z1_GPREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPREG3	R	0h	This field gets loaded with the contents of Z1OTP_GPREG3 locations in Zone-1-USER-OTP when a dummy read is issued to that address. Users can use this register to load any general purpose non-voliate content from USER-OTP of Zone-1 Reset type: SYSRSn

5.9.2.8 Z1_GPREG4 Register (Offset = Eh) [Reset = 00000000h]

Z1_GPREG4 is shown in Figure 5-12 and described in Table 5-14.

Return to the Summary Table.

Zone 1 General Purpose Register-4

Figure 5-12. Z1_GPREG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPREG4																															
R-0h																															

Table 5-14. Z1_GPREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPREG4	R	0h	This field gets loaded with the contents of Z1OTP_GPREG4 locations in Zone-1-USER-OTP when a dummy read is issued to that address. Users can use this register to load any general purpose non-voliate content from USER-OTP of Zone-1 Reset type: SYSRSn

5.9.2.9 Z1_CSMKEY0 Register (Offset = 10h) [Reset = 00000000h]

Z1_CSMKEY0 is shown in Figure 5-13 and described in Table 5-15.

Return to the Summary Table.

Zone 1 CSM Key 0

Figure 5-13. Z1_CSMKEY0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1_CSMKEY0																															
R/W-0h																															

Table 5-15. Z1_CSMKEY0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1_CSMKEY0	R/W	0h	To unlock Zone1, user needs to write this regsiter with exact value as Z1_CSMPSWD0, programmed in OTP (zone gets unlock only if 128 bit password in OTP match with value written in four CSMKEY registers.) Reset type: SYSRSn

5.9.2.10 Z1_CSMKEY1 Register (Offset = 12h) [Reset = 00000000h]

Z1_CSMKEY1 is shown in Figure 5-14 and described in Table 5-16.

Return to the Summary Table.

Zone 1 CSM Key 1

Figure 5-14. Z1_CSMKEY1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1_CSMKEY1																															
R/W-0h																															

Table 5-16. Z1_CSMKEY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1_CSMKEY1	R/W	0h	To unlock Zone1, user needs to write this regsiter with exact value as Z1_CSMPSWD1, programmed in OTP (zone gets unlock only if 128 bit password in OTP match with value written in four CSMKEY registers.) Reset type: SYSRSn

5.9.2.11 Z1_CSMKEY2 Register (Offset = 14h) [Reset = 00000000h]

Z1_CSMKEY2 is shown in Figure 5-15 and described in Table 5-17.

Return to the Summary Table.

Zone 1 CSM Key 2

Figure 5-15. Z1_CSMKEY2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1_CSMKEY2																															
R/W-0h																															

Table 5-17. Z1_CSMKEY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1_CSMKEY2	R/W	0h	To unlock Zone1, user needs to write this regsiter with exact value as Z1_CSMPSWD2, programmed in OTP (zone gets unlock only if 128 bit password in OTP match with value written in four CSMKEY registers.) Reset type: SYSRSn

5.9.2.12 Z1_CSMKEY3 Register (Offset = 16h) [Reset = 00000000h]

Z1_CSMKEY3 is shown in Figure 5-16 and described in Table 5-18.

Return to the Summary Table.

Zone 1 CSM Key 3

Figure 5-16. Z1_CSMKEY3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1_CSMKEY3																															
R/W-0h																															

Table 5-18. Z1_CSMKEY3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1_CSMKEY3	R/W	0h	To unlock Zone1, user needs to write this regsiter with exact value as Z1_CSMPSWD3, programmed in OTP (zone gets unlock only if 128 bit password in OTP match with value written in four CSMKEY registers.) Reset type: SYSRSn

5.9.2.13 Z1_CR Register (Offset = 18h) [Reset = 00080000h]

Z1_CR is shown in Figure 5-17 and described in Table 5-19.

Return to the Summary Table.

Zone 1 CSM Control Register

Figure 5-17. Z1_CR Register

31	30	29	28	27	26	25	24
FORCESEC	RESERVED						
R-0/W-0h	R-0h						
23	22	21	20	19	18	17	16
RESERVED	ARMED	UNSECURE	ALLONE	ALLZERO	RESERVED		
R-0h	R-0h	R-0h	R-0h	R-1h	R-0h		
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h				R-0h	R-0h	R-0h	R-0h

Table 5-19. Z1_CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	FORCESEC	R-0/W	0h	A write of '1' to this bit clears the CSMKEYx registers. If writing to this bit after performing an update of the security passwords, it is advised to immediately perform a dummy load of the passwords by initiating a read of the passwords from their flash locations. Reset type: SYSRSn
30-24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	ARMED	R	0h	0 : Dummy read to CSM Password locations in OTP hasn't been performed. 1 : Dummy read to CSM Password locations in OTP has been performed. Reset type: SYSRSn
21	UNSECURE	R	0h	Indiacates the state of Zone. 0 : Zone is in lock(secure) state. 1 : Zone is in unlock(unsecure) state. Reset type: SYSRSn
20	ALLONE	R	0h	Indicates the state of CSM passwords. 0 : Zone CSM Passwords are not all ones. 1 : Zone CSM Passwords are all ones. Reset type: SYSRSn
19	ALLZERO	R	1h	Indicates the state of CSM passowrds. 0 : CSM Passwords are not all zeros. 1 : CSM Passwords are all zero and device is permanently locked. Reset type: SYSRSn
18-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved

Table 5-19. Z1_CR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RESERVED	R	0h	Reserved

5.9.2.14 Z1_GRABSECT1R Register (Offset = 1Ah) [Reset = 00000000h]

Z1_GRABSECT1R is shown in Figure 5-18 and described in Table 5-20.

Return to the Summary Table.

Zone 1 Grab Flash Status Register 1

Figure 5-18. Z1_GRABSECT1R Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				GRAB_B0_SECT63_32		GRAB_B0_SECT31_4	
R-0h				R-0h		R-0h	
7	6	5	4	3	2	1	0
GRAB_B0_SECT3		GRAB_B0_SECT2		GRAB_B0_SECT1		GRAB_B0_SECT0	
R-0h		R-0h		R-0h		R-0h	

Table 5-20. Z1_GRABSECT1R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-10	GRAB_B0_SECT63_32	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z1_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone1. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn
9-8	GRAB_B0_SECT31_4	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z1_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone1. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn
7-6	GRAB_B0_SECT3	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z1_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone1. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn

Table 5-20. Z1_GRABSECT1R Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	GRAB_B0_SECT2	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z1_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone1. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn
3-2	GRAB_B0_SECT1	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z1_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone1. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn
1-0	GRAB_B0_SECT0	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z1_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone1. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn

5.9.2.15 Z1_EXEONLYSECT1R Register (Offset = 26h) [Reset = 00000000h]

Z1_EXEONLYSECT1R is shown in Figure 5-19 and described in Table 5-21.

Return to the Summary Table.

Zone 1 Execute Only Flash Status Register 1

Figure 5-19. Z1_EXEONLYSECT1R Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		EXEONLY_B0_SECT63_32	EXEONLY_B0_SECT31_4	EXEONLY_B0_SECT3	EXEONLY_B0_SECT2	EXEONLY_B0_SECT1	EXEONLY_B0_SECT0
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 5-21. Z1_EXEONLYSECT1R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	EXEONLY_B0_SECT63_32	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z1_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone1) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone1) Reset type: SYSRSn
4	EXEONLY_B0_SECT31_4	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z1_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone1) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone1) Reset type: SYSRSn
3	EXEONLY_B0_SECT3	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z1_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone1) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone1) Reset type: SYSRSn

Table 5-21. Z1_EXEONLYSECT1R Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	EXEONLY_B0_SECT2	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z1_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone1) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone1) Reset type: SYSRSn
1	EXEONLY_B0_SECT1	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z1_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone1) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone1) Reset type: SYSRSn
0	EXEONLY_B0_SECT0	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z1_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone1) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone1) Reset type: SYSRSn

5.9.2.16 Z1_JTAGKEY0 Register (Offset = 2Eh) [Reset = 00000000h]

Z1_JTAGKEY0 is shown in Figure 5-20 and described in Table 5-22.

Return to the Summary Table.

JTAG Unlock Key Register 0

Figure 5-20. Z1_JTAGKEY0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0																															
R-0h																															

Table 5-22. Z1_JTAGKEY0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY0	R	0h	Value in this field is scanned in via the JLM scan chain. JTAGKEY is compared to a hidden register that gets dummy loaded when a read is issued to the JTAGPSWD locations in OTP. Reset type: PORESETn

5.9.2.17 Z1_JTAGKEY1 Register (Offset = 30h) [Reset = 00000000h]

Z1_JTAGKEY1 is shown in Figure 5-21 and described in Table 5-23.

Return to the Summary Table.

JTAG Unlock Key Register 1

Figure 5-21. Z1_JTAGKEY1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1																															
R-0h																															

Table 5-23. Z1_JTAGKEY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY1	R	0h	Value in this field is scanned in via the JLM scan chain. JTAGKEY is compared to a hidden register that gets dummy loaded when a read is issued to the JTAGPSWD locations in OTP. Reset type: PORESETn

5.9.2.18 Z1_JTAGKEY2 Register (Offset = 32h) [Reset = 00000000h]

Z1_JTAGKEY2 is shown in Figure 5-22 and described in Table 5-24.

Return to the Summary Table.

JTAG Unlock Key Register 2

Figure 5-22. Z1_JTAGKEY2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2																															
R-0h																															

Table 5-24. Z1_JTAGKEY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY2	R	0h	Value in this field is scanned in via the JLM scan chain. JTAGKEY is compared to a hidden register that gets dummy loaded when a read is issued to the JTAGPSWD locations in OTP. Reset type: PORESETn

5.9.2.19 Z1_JTAGKEY3 Register (Offset = 34h) [Reset = 00000000h]

Z1_JTAGKEY3 is shown in Figure 5-23 and described in Table 5-25.

Return to the Summary Table.

JTAG Unlock Key Register 3

Figure 5-23. Z1_JTAGKEY3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3																															
R-0h																															

Table 5-25. Z1_JTAGKEY3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY3	R	0h	Value in this field is scanned in via the JLM scan chain. JTAGKEY is compared to a hidden register that gets dummy loaded when a read is issued to the JTAGPSWD locations in OTP. Reset type: PORESETn

5.9.2.20 Z1_CMACKEY0 Register (Offset = 36h) [Reset = 00000000h]

Z1_CMACKEY0 is shown in Figure 5-24 and described in Table 5-26.

Return to the Summary Table.

Secure Boot CMAC Key Status Register 0

Figure 5-24. Z1_CMACKEY0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0																															
R-0h																															

Table 5-26. Z1_CMACKEY0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY0	R	0h	Value in this field gets loaded from CMACKEY0 when a read is issued to its address in OTP. Reset type: SYSRSn

5.9.2.21 Z1_CMACKEY1 Register (Offset = 38h) [Reset = 00000000h]

Z1_CMACKEY1 is shown in Figure 5-25 and described in Table 5-27.

Return to the Summary Table.

Secure Boot CMAC Key Status Register 1

Figure 5-25. Z1_CMACKEY1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1																															
R-0h																															

Table 5-27. Z1_CMACKEY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY1	R	0h	Value in this field gets loaded from CMACKEY1 when a read is issued to its address in OTP. Reset type: SYSRSn

5.9.2.22 Z1_CMACKEY2 Register (Offset = 3Ah) [Reset = 00000000h]

Z1_CMACKEY2 is shown in Figure 5-26 and described in Table 5-28.

Return to the Summary Table.

Secure Boot CMAC Key Status Register 2

Figure 5-26. Z1_CMACKEY2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2																															
R-0h																															

Table 5-28. Z1_CMACKEY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY2	R	0h	Value in this field gets loaded from CMACKEY2 when a read is issued to its address in OTP. Reset type: SYSRSn

5.9.2.23 Z1_CMACKEY3 Register (Offset = 3Ch) [Reset = 00000000h]

Z1_CMACKEY3 is shown in Figure 5-27 and described in Table 5-29.

Return to the Summary Table.

Secure Boot CMAC Key Status Register 3

Figure 5-27. Z1_CMACKEY3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3																															
R-0h																															

Table 5-29. Z1_CMACKEY3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	KEY3	R	0h	Value in this field gets loaded from CMACKEY3 when a read is issued to its address in OTP. Reset type: SYSRSn

5.9.3 DCSM_Z2_REGS Registers

Table 5-30 lists the memory-mapped registers for the DCSM_Z2_REGS registers. All register offset addresses not listed in Table 5-30 should be considered as reserved locations and the register contents should not be modified.

Table 5-30. DCSM_Z2_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	Z2_LINKPOINTER	Zone 2 Link Pointer	
2h	Z2_OTPSECLOCK	Zone 2 OTP Secure Lock	
6h	Z2_LINKPOINTERERR	Link Pointer Error	
8h	Z2_GPREG1	Zone 2 General Purpose Register-1	
Ah	Z2_GPREG2	Zone 2 General Purpose Register-2	
Ch	Z2_GPREG3	Zone 2 General Purpose Register-3	
Eh	Z2_GPREG4	Zone 2 General Purpose Register-4	
10h	Z2_CSMKEY0	Zone 2 CSM Key 0	
12h	Z2_CSMKEY1	Zone 2 CSM Key 1	
14h	Z2_CSMKEY2	Zone 2 CSM Key 2	
16h	Z2_CSMKEY3	Zone 2 CSM Key 3	
18h	Z2_CR	Zone 2 CSM Control Register	
1Ah	Z2_GRABSECT1R	Zone 2 Grab Flash Status Register 1	
26h	Z2_EXEONLYSECT1R	Zone 2 Execute Only Flash Status Register 1	

Complex bit access types are encoded to fit into small table cells. Table 5-31 shows the codes that are used for access types in this section.

Table 5-31. DCSM_Z2_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

5.9.3.1 Z2_LINKPOINTER Register (Offset = 0h) [Reset = FFFFC000h]

Z2_LINKPOINTER is shown in Figure 5-28 and described in Table 5-32.

Return to the Summary Table.

Zone 2 Link Pointer

Figure 5-28. Z2_LINKPOINTER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		LINKPOINTER													
R-0003FFFFh																		R-0h													

Table 5-32. Z2_LINKPOINTER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0003FFFFh	Reserved
13-0	LINKPOINTER	R	0h	This is resolved Link-Pointer value which is generated by looking at the three physical Link-Pointer values loaded from OTP Reset type: SYSRSn

5.9.3.2 Z2_OTPSECLOCK Register (Offset = 2h) [Reset = 00000001h]

Z2_OTPSECLOCK is shown in Figure 5-29 and described in Table 5-33.

Return to the Summary Table.

Zone 2 OTP Secure Lock

Figure 5-29. Z2_OTPSECLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				RESERVED			
R-0h				R-0h			
7	6	5	4	3	2	1	0
PSWDLOCK				RESERVED			JTAGLOCK
R-0h				R-0h			R-1h

Table 5-33. Z2_OTPSECLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	RESERVED	R	0h	Reserved
11-8	RESERVED	R	0h	Reserved
7-4	PSWDLOCK	R	0h	Value in this field gets loaded from Z2_PSWDLOCK[3:0] when a read is issued to address location of Z2_PSWDLOCK in OTP. 1111 : CSM password locations in OTP are not protected and can be read from debugger as well as code running from anywhere. Other Value : CSM password locations in OTP are protected and can't be read without unlocking CSM of that zone. Reset type: XRSn
3-1	RESERVED	R	0h	Reserved
0	JTAGLOCK	R	1h	Reflects the state of the JTAGLOCK feature. 0 : JTAG is not locked 1 : JTAG is locked This bit is a copy of the Z1_OTPSECLOCK.JTAGLOCK bit. Reset type: PORESETn

5.9.3.3 Z2_LINKPOINTERERR Register (Offset = 6h) [Reset = 00000000h]

Z2_LINKPOINTERERR is shown in Figure 5-30 and described in Table 5-34.

Return to the Summary Table.

Link Pointer Error

Figure 5-30. Z2_LINKPOINTERERR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		Z2_LINKPOINTERERR													
R-0-0h		R-0h													

Table 5-34. Z2_LINKPOINTERERR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	Z2_LINKPOINTERERR	R	0h	These bits indicate errors during formation of the resolved Link-Pointer value after the three physical Link-Pointer values loaded of OTP in flash. 0 : No Error. Other : Error on bit positions which is set to 1. Reset type: SYSRSn

5.9.3.4 Z2_GPREG1 Register (Offset = 8h) [Reset = 00000000h]

Z2_GPREG1 is shown in Figure 5-31 and described in Table 5-35.

Return to the Summary Table.

Zone 2 General Purpose Register-1

Figure 5-31. Z2_GPREG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPREG1																															
R-0h																															

Table 5-35. Z2_GPREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPREG1	R	0h	This field gets loaded with the contents of Z2OTP_GPREG1 locations in Zone-2-USER-OTP when a dummy read is issued to that address. Users can use this register to load any general purpose non-voliate content from USER-OTP of Zone-2 Reset type: SYSRSn

5.9.3.5 Z2_GPREG2 Register (Offset = Ah) [Reset = 00000000h]

Z2_GPREG2 is shown in Figure 5-32 and described in Table 5-36.

Return to the Summary Table.

Zone 2 General Purpose Register-2

Figure 5-32. Z2_GPREG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPREG2																															
R-0h																															

Table 5-36. Z2_GPREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPREG2	R	0h	This field gets loaded with the contents of Z2OTP_GPREG2 locations in Zone-2-USER-OTP when a dummy read is issued to that address. Users can use this register to load any general purpose non-voliate content from USER-OTP of Zone-2 Reset type: SYSRSn

5.9.3.6 Z2_GPREG3 Register (Offset = Ch) [Reset = 00000000h]

Z2_GPREG3 is shown in Figure 5-33 and described in Table 5-37.

Return to the Summary Table.

Zone 2 General Purpose Register-3

Figure 5-33. Z2_GPREG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPREG3																															
R-0h																															

Table 5-37. Z2_GPREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPREG3	R	0h	This field gets loaded with the contents of Z2OTP_GPREG3 locations in Zone-2-USER-OTP when a dummy read is issued to that address. Users can use this register to load any general purpose non-voliate content from USER-OTP of Zone-2 Reset type: SYSRSn

5.9.3.7 Z2_GPREG4 Register (Offset = Eh) [Reset = 00000000h]

Z2_GPREG4 is shown in Figure 5-34 and described in Table 5-38.

Return to the Summary Table.

Zone 2 General Purpose Register-4

Figure 5-34. Z2_GPREG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPREG4																															
R-0h																															

Table 5-38. Z2_GPREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPREG4	R	0h	This field gets loaded with the contents of Z2OTP_GPREG4 locations in Zone-2-USER-OTP when a dummy read is issued to that address. Users can use this register to load any general purpose non-voliate content from USER-OTP of Zone-2 Reset type: SYSRSn

5.9.3.8 Z2_CSMKEY0 Register (Offset = 10h) [Reset = 00000000h]

Z2_CSMKEY0 is shown in Figure 5-35 and described in Table 5-39.

Return to the Summary Table.

Zone 2 CSM Key 0

Figure 5-35. Z2_CSMKEY0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2_CSMKEY0																															
R/W-0h																															

Table 5-39. Z2_CSMKEY0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2_CSMKEY0	R/W	0h	To unlock Zone2, user needs to write this regsiter with exact value as Z2_CSMPSWD0, programmed in OTP (zone gets unlock only if 128 bit password in OTP match with value written in four CSMKEY registers.) Reset type: SYSRSn

5.9.3.9 Z2_CSMKEY1 Register (Offset = 12h) [Reset = 00000000h]

Z2_CSMKEY1 is shown in Figure 5-36 and described in Table 5-40.

Return to the Summary Table.

Zone 2 CSM Key 1

Figure 5-36. Z2_CSMKEY1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2_CSMKEY1																															
R/W-0h																															

Table 5-40. Z2_CSMKEY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2_CSMKEY1	R/W	0h	To unlock Zone2, user needs to write this regsiter with exact value as Z2_CSMPSWD1, programmed in OTP (zone gets unlock only if 128 bit password in OTP match with value written in four CSMKEY registers.) Reset type: SYSRSn

5.9.3.10 Z2_CSMKEY2 Register (Offset = 14h) [Reset = 00000000h]

Z2_CSMKEY2 is shown in Figure 5-37 and described in Table 5-41.

Return to the Summary Table.

Zone 2 CSM Key 2

Figure 5-37. Z2_CSMKEY2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2_CSMKEY2																															
R/W-0h																															

Table 5-41. Z2_CSMKEY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2_CSMKEY2	R/W	0h	To unlock Zone2, user needs to write this regsiter with exact value as Z2_CSMPSWD2, programmed in OTP (zone gets unlock only if 128 bit password in OTP match with value written in four CSMKEY registers.) Reset type: SYSRSn

5.9.3.11 Z2_CSMKEY3 Register (Offset = 16h) [Reset = 00000000h]

Z2_CSMKEY3 is shown in Figure 5-38 and described in Table 5-42.

Return to the Summary Table.

Zone 2 CSM Key 3

Figure 5-38. Z2_CSMKEY3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2_CSMKEY3																															
R/W-0h																															

Table 5-42. Z2_CSMKEY3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2_CSMKEY3	R/W	0h	To unlock Zone2, user needs to write this regsiter with exact value as Z2_CSMPSWD3, programmed in OTP (zone gets unlock only if 128 bit password in OTP match with value written in four CSMKEY registers.) Reset type: SYSRSn

5.9.3.12 Z2_CR Register (Offset = 18h) [Reset = 00080000h]

Z2_CR is shown in Figure 5-39 and described in Table 5-43.

Return to the Summary Table.

Zone 2 CSM Control Register

Figure 5-39. Z2_CR Register

31	30	29	28	27	26	25	24
FORCESEC	RESERVED						
R-0/W-0h	R-0h						
23	22	21	20	19	18	17	16
RESERVED	ARMED	UNSECURE	ALLONE	ALLZERO	RESERVED		
R-0h	R-0h	R-0h	R-0h	R-1h	R-0h		
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h				R-0h	R-0h	R-0h	R-0h

Table 5-43. Z2_CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	FORCESEC	R-0/W	0h	A write of '1' to this bit clears the CSMKEYx registers. If writing to this bit after performing an update of the security passwords, it is advised to immediately perform a dummy load of the passwords by initiating a read of the passwords from their flash locations. Reset type: SYSRSn
30-24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	ARMED	R	0h	0 : Dummy read to CSM Password locations in OTP hasn't been performed. 1 : Dummy read to CSM Password locations in OTP has been performed. Reset type: SYSRSn
21	UNSECURE	R	0h	Indiacates the state of Zone. 0 : Zone is in lock(secure) state. 1 : Zone is in unlock(unsecure) state. Reset type: SYSRSn
20	ALLONE	R	0h	Indicates the state of CSM passwords. 0 : Zone CSM Passwords are not all ones. 1 : Zone CSM Passwords are all ones. Reset type: SYSRSn
19	ALLZERO	R	1h	Indicates the state of CSM passowrds. 0 : CSM Passwords are not all zeros. 1 : CSM Passwords are all zero and device is permanently locked. Reset type: SYSRSn
18-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved

Table 5-43. Z2_CR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RESERVED	R	0h	Reserved

5.9.3.13 Z2_GRABSECT1R Register (Offset = 1Ah) [Reset = 00000000h]

Z2_GRABSECT1R is shown in Figure 5-40 and described in Table 5-44.

Return to the Summary Table.

Zone 2 Grab Flash Status Register 1

Figure 5-40. Z2_GRABSECT1R Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				GRAB_B0_SECT63_32		GRAB_B0_SECT31_4	
R-0h				R-0h		R-0h	
7	6	5	4	3	2	1	0
GRAB_B0_SECT3		GRAB_B0_SECT2		GRAB_B0_SECT1		GRAB_B0_SECT0	
R-0h		R-0h		R-0h		R-0h	

Table 5-44. Z2_GRABSECT1R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-10	GRAB_B0_SECT63_32	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z2_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone2. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn
9-8	GRAB_B0_SECT31_4	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z2_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone2. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn
7-6	GRAB_B0_SECT3	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z2_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone2. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn

Table 5-44. Z2_GRABSECT1R Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	GRAB_B0_SECT2	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z2_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone2. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn
3-2	GRAB_B0_SECT1	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z2_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone2. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn
1-0	GRAB_B0_SECT0	R	0h	Value in this field gets loaded from the equivalent bit field when a read is issued to the active ZSB address location of Z2_GRABSECT1 in the SECURITY sector. 00 : Invalid. Sectors are inaccessible. 01 : Request to allocate thes sectors to Zone2. 10 : No request for these sectors. 11 : No request for these sectors when this zone is UNLOCKED. Else these sectors are inaccessible if this zone is LOCKED. Reset type: SYSRSn

5.9.3.14 Z2_EXEONLYSECT1R Register (Offset = 26h) [Reset = 00000000h]

Z2_EXEONLYSECT1R is shown in Figure 5-41 and described in Table 5-45.

Return to the Summary Table.

Zone 2 Execute Only Flash Status Register 1

Figure 5-41. Z2_EXEONLYSECT1R Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		EXEONLY_B0_SECT63_32	EXEONLY_B0_SECT31_4	EXEONLY_B0_SECT3	EXEONLY_B0_SECT2	EXEONLY_B0_SECT1	EXEONLY_B0_SECT0
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 5-45. Z2_EXEONLYSECT1R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	EXEONLY_B0_SECT63_32	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z2_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone2) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone2) Reset type: SYSRSn
4	EXEONLY_B0_SECT31_4	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z2_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone2) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone2) Reset type: SYSRSn
3	EXEONLY_B0_SECT3	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z2_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone2) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone2) Reset type: SYSRSn

Table 5-45. Z2_EXEONLYSECT1R Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	EXEONLY_B0_SECT2	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z2_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone2) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone2) Reset type: SYSRSn
1	EXEONLY_B0_SECT1	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z2_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone2) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone2) Reset type: SYSRSn
0	EXEONLY_B0_SECT0	R	0h	Value in this bit gets loaded from the equivalent bit when a read is issued to the Z2_EXEONLYSECT1 address location in the SECURITY sector. 0 : Execute-Only protection is enabled for this sector (only if it is allocated to Zone2) 1 : Execute-Only protection is disabled for this sector (only if it is allocated to Zone2T) Reset type: SYSRSn

5.9.4 DCSM_COMMON_REGS Registers

Table 5-46 lists the memory-mapped registers for the DCSM_COMMON_REGS registers. All register offset addresses not listed in Table 5-46 should be considered as reserved locations and the register contents should not be modified.

Table 5-46. DCSM_COMMON_REGS Registers

Offset	Acronym	Register Name	Write Protection
8h	SECTSTAT1	Flash Sectors Status Register 1	
18h	SECERRSTAT	Security Error Status Register	
1Ah	SECERRCLR	Security Error Clear Register	
1Ch	SECERRFRC	Security Error Force Register	
1Eh	DENYCODE	Flash Authorization Denial Code	
28h	UID_UNIQUE_31_0	Unique Identification Number Low	
2Ah	UID_UNIQUE_63_32	Unique Identification Number High	
2Ch	PARTIDH	Part Identification High Register	
2Eh	PERSEM1	Peripheral Semaphore Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 5-47 shows the codes that are used for access types in this section.

Table 5-47. DCSM_COMMON_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WOnce	W Once	Write Write once
Reset or Default Value		
-n		Value after reset or the default value

5.9.4.1 SECTSTAT1 Register (Offset = 8h) [Reset = 00000000h]

SECTSTAT1 is shown in Figure 5-42 and described in Table 5-48.

Return to the Summary Table.

Flash Sectors Status Register 1

Figure 5-42. SECTSTAT1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				STATUS_B0_SECT63_32		STATUS_B0_SECT31_4	
R-0h				R-0h		R-0h	
7	6	5	4	3	2	1	0
STATUS_B0_SECT3		STATUS_B0_SECT2		STATUS_B0_SECT1		STATUS_B0_SECT0	
R-0h		R-0h		R-0h		R-0h	

Table 5-48. SECTSTAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-10	STATUS_B0_SECT63_32	R	0h	Reflects the status of given flash sector. 00 : Sector is in-accessible 01 : Sector belongs to Zone1. 10 : Sector belongs to Zone2. 11: Sector is un-secure and code running in both zone have full access to it. Reset type: SYSRSn
9-8	STATUS_B0_SECT31_4	R	0h	Reflects the status of given flash sector. 00 : Sector is in-accessible 01 : Sector belongs to Zone1. 10 : Sector belongs to Zone2. 11: Sector is un-secure and code running in both zone have full access to it. Reset type: SYSRSn
7-6	STATUS_B0_SECT3	R	0h	Reflects the status of given flash sector. 00 : Sector is in-accessible 01 : Sector belongs to Zone1. 10 : Sector belongs to Zone2. 11: Sector is un-secure and code running in both zone have full access to it. Reset type: SYSRSn
5-4	STATUS_B0_SECT2	R	0h	Reflects the status of given flash sector. 00 : Sector is in-accessible 01 : Sector belongs to Zone1. 10 : Sector belongs to Zone2. 11: Sector is un-secure and code running in both zone have full access to it. Reset type: SYSRSn

Table 5-48. SECTSTAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	STATUS_B0_SECT1	R	0h	Reflects the status of given flash sector. 00 : Sector is in-accessible 01 : Sector belongs to Zone1. 10 : Sector belongs to Zone2. 11: Sector is un-secure and code running in both zone have full access to it. Reset type: SYSRSn
1-0	STATUS_B0_SECT0	R	0h	Reflects the status of given flash sector. 00 : Sector is in-accessible 01 : Sector belongs to Zone1. 10 : Sector belongs to Zone2. 11: Sector is un-secure and code running in both zone have full access to it. Reset type: SYSRSn

5.9.4.2 SECERRSTAT Register (Offset = 18h) [Reset = 00000000h]

SECERRSTAT is shown in Figure 5-43 and described in Table 5-49.

Return to the Summary Table.

Security Error Status Register

Figure 5-43. SECERRSTAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ERR
R-0-0h															R-0h

Table 5-49. SECERRSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	ERR	R	0h	This bit indicates if any error has occurred in the load of any security configuration from USER-OTP. 0: No error has occurred in the load of security information from USER-OTP 1: Error has occurred in the load of security information from USER-OTP Reset type: PORESETn

5.9.4.3 SECERRCLR Register (Offset = 1Ah) [Reset = 00000000h]

SECERRCLR is shown in Figure 5-44 and described in Table 5-50.

Return to the Summary Table.

Security Error Clear Register

Figure 5-44. SECERRCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ERR
R-0-0h															R-0/W1S-0h

Table 5-50. SECERRCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	ERR	R-0/W1S	0h	A write of '1' clears the SECERRSTAT.ERR bit. Write of '0' is ignored. This bit always reads back '0'. Reset type: N/A

5.9.4.4 SECERRFRC Register (Offset = 1Ch) [Reset = 00000000h]

SECERRFRC is shown in Figure 5-45 and described in Table 5-51.

Return to the Summary Table.

Security Error Force Register

Figure 5-45. SECERRFRC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
R-0/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ERR
R-0-0h															R-0/W1S-0h

Table 5-51. SECERRFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	In order to write to the ERR bits, 0x5a5a must be written to these key bits at the same time. Otherwise, writes are ignored. The key is cleared immediately after writing, so it must be written again for every write to ERR. Reads will return 0. Reset type: N/A
15-1	RESERVED	R-0	0h	Reserved
0	ERR	R-0/W1S	0h	A write of '1', along with the proper KEY, sets the SECERRSTAT.ERR bit. Write of '0' is ignored. This bit always reads back '0'. Reset type: N/A

5.9.4.5 DENYCODE Register (Offset = 1Eh) [Reset = 00000000h]

DENYCODE is shown in Figure 5-46 and described in Table 5-52.

Return to the Summary Table.

Flash Authorization Denial Code

Figure 5-46. DENYCODE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
ILLSIZE	ILLCMD	ILLMODECH	ILLRDVER	ILLERASE	ILLPROG	ILLADDR	BLOCKED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 5-52. DENYCODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7	ILLSIZE	R	0h	This bit indicates the DCSM stopped a Flash Controller operation because an illegal command size was requested. This bit is not sticky. It is updated each time a Flash Controller operation is denied. 0 : Flash operation was not stopped due to an illegal command size 1 : Flash operation was stopped due to an illegal command size Reset type: SYSRSn
6	ILLCMD	R	0h	This bit indicates the DCSM stopped a Flash Controller operation because an illegal command type was requested. This bit is not sticky. It is updated each time a Flash Controller operation is denied. 0 : Flash operation was not stopped due to an illegal command type 1 : Flash operation was stopped due to an illegal command type Reset type: SYSRSn
5	ILLMODECH	R	0h	This bit indicates the DCSM stopped a Flash Controller operation because a mode change command tried to move it into a reserved mode. This bit is not sticky. It is updated each time a Flash Controller operation is denied. 0 : Flash operation was not stopped due to an illegal mode change 1 : Flash operation was stopped due to an illegal mode change Reset type: SYSRSn
4	ILLRDVER	R	0h	This bit indicates the DCSM stopped a Flash Controller operation because a read verify command provided an illegal address. This bit is not sticky. It is updated each time a Flash Controller operation is denied. 0 : Flash operation was not stopped due to an illegal read verify address 1 : Flash operation was stopped due to an illegal read verify address Reset type: SYSRSn

Table 5-52. DENYCODE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	ILLERASE	R	0h	This bit indicates the DCSM stopped a Flash Controller operation because an erase command provided an illegal address. This bit is not sticky. It is updated each time a Flash Controller operation is denied. 0 : Flash operation was not stopped due to an illegal erase address 1 : Flash operation was stopped due to an illegal erase address Reset type: SYSRSn
2	ILLPROG	R	0h	This bit indicates the DCSM stopped a Flash Controller operation because a programming command provided an illegal address. This bit is not sticky. It is updated each time a Flash Controller operation is denied. 0 : Flash operation was not stopped due to an illegal programming address 1 : Flash operation was stopped due to an illegal programming address Reset type: SYSRSn
1	ILLADDR	R	0h	This bit indicates the DCSM stopped a Flash Controller operation because the command provided contained a non-flash address. This bit is not sticky. It is updated each time a Flash Controller operation is denied. 0 : Flash operation was not stopped due to a non-flash address 1 : Flash operation was stopped due to a non-flash address Reset type: SYSRSn
0	BLOCKED	R	0h	This bit indicates the DCSM stopped a Flash Controller operation because the DCSM was in the BLOCKED state. This bit is not sticky. It is updated each time a Flash Controller operation is denied. 0 : Flash operation was not stopped due to the BLOCKED state 1 : Flash operation was stopped due to the BLOCKED state Reset type: SYSRSn

5.9.4.6 UID_UNIQUE_31_0 Register (Offset = 28h) [Reset = 00000000h]

UID_UNIQUE_31_0 is shown in Figure 5-47 and described in Table 5-53.

Return to the Summary Table.

Unique Identification Number Low

Figure 5-47. UID_UNIQUE_31_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID_L																															
R/WOnce-0h																															

Table 5-53. UID_UNIQUE_31_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID_L	R/WOnce	0h	This register contains a copy of the device UID_UNIQUE value bits 31 to 0. Reset type: PORESETn

5.9.4.7 UID_UNIQUE_63_32 Register (Offset = 2Ah) [Reset = 00000000h]

UID_UNIQUE_63_32 is shown in Figure 5-48 and described in Table 5-54.

Return to the Summary Table.

Unique Identification Number High

Figure 5-48. UID_UNIQUE_63_32 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID_H																															
R/WOnce-0h																															

Table 5-54. UID_UNIQUE_63_32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	UID_H	R/WOnce	0h	This register contains a copy of the device UID_UNIQUE value bits 63 to 32. Reset type: PORESETn

5.9.4.8 PARTIDH Register (Offset = 2Ch) [Reset = 00000000h]

PARTIDH is shown in Figure 5-49 and described in Table 5-55.

Return to the Summary Table.

Part Identification High Register

Figure 5-49. PARTIDH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																															
R/WOnce-0h																															

Table 5-55. PARTIDH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ID	R/WOnce	0h	This register contains a copy of the device PARTIDH value. Reset type: PORESETn

5.9.4.9 PERSEM1 Register (Offset = 2Eh) [Reset = 00000000h]

PERSEM1 is shown in Figure 5-50 and described in Table 5-56.

Return to the Summary Table.

Peripheral Semaphore Register

Figure 5-50. PERSEM1 Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED						GRABRSTCTL	
R-0-0h						R/W-0h	
7	6	5	4	3	2	1	0
GRABCLKCTL		GRABTIMER1		GRABNMIWD		GRABWD	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 5-56. PERSEM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	Writing a value 0xA5 into this field will allow the update to any bit field, else writes are ignored. Reads will return 0. Reset type: SYSRSn
23-16	RESERVED	R-0	0h	Reserved
15-10	RESERVED	R-0	0h	Reserved
9-8	GRABRSTCTL	R/W	0h	Grab Reset configuration. Reset type: SYSRSn
7-6	GRABCLKCTL	R/W	0h	Grab Clock configuration. Reset type: SYSRSn
5-4	GRABTIMER1	R/W	0h	Grab TIMER1 module. Reset type: SYSRSn
3-2	GRABNMIWD	R/W	0h	GRAB NMIWD module. Reset type: SYSRSn

Table 5-56. PERSEM1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	GRABWD	R/W	0h	Grab Watchdog module 00 : Module configuration registers can be written by code running from anywhere without any restriction. 01 : Module configuration registers can be written by code running from Zone1 security zone. 10 : Module configuration registers can be written by code running from Zone2 security zone 11 : Module configuration registers can be written by code running from anywhere without any restriction Allowed State Transitions in this field. 00 TO 11 : Not allowed. 11 TO 00 : Not allowed. 00/11 TO 01 : Code running from Zone1 only can perform this transition. 01 TO 00/11 : Code running from Zone1 only can perform this transition. 00/11 TO 10 : Code running from Zone2 only can perform this transition. 10 TO 00/11 : Code running from Zone2 can perform this transition 10 TO 01 : Not allowed. 01 TO 10 : Not allowed. Reset type: SYSRSn

5.9.5 DCSM_Z1_OTP Registers

Table 5-57 lists the memory-mapped registers for the DCSM_Z1_OTP registers. All register offset addresses not listed in Table 5-57 should be considered as reserved locations and the register contents should not be modified.

Table 5-57. DCSM_Z1_OTP Registers

Offset	Acronym	Register Name	Write Protection
0h	Z1OTP_LINKPOINTER1	Zone 1 Link Pointer1	
2h	Z1OTP_LINKPOINTER2	Zone 1 Link Pointer2	
4h	Z1OTP_LINKPOINTER3	Zone 1 Link Pointer3	
6h	Z1OTP_JLM_ENABLE	Zone 1 JTAGLOCK Enable Register	
8h	Z1OTP_GPREG1	Zone 1 General Purpose Register 1	
Ah	Z1OTP_GPREG2	Zone 1 General Purpose Register 2	
Ch	Z1OTP_GPREG3	Zone 1 General Purpose Register 3	
Eh	Z1OTP_GPREG4	Zone 1 General Purpose Register 4	
10h	Z1OTP_PSWDLOCK	Secure Password Lock	
14h	Z1OTP_JTAGPSWDH0	JTAG Lock Permanent Password 0	
16h	Z1OTP_JTAGPSWDH1	JTAG Lock Permanent Password 1	
18h	Z1OTP_CMACKEY0	Secure Boot CMAC Key 0	
1Ah	Z1OTP_CMACKEY1	Secure Boot CMAC Key 1	
1Ch	Z1OTP_CMACKEY2	Secure Boot CMAC Key 2	
1Eh	Z1OTP_CMACKEY3	Secure Boot CMAC Key 3	

Complex bit access types are encoded to fit into small table cells. Table 5-58 shows the codes that are used for access types in this section.

Table 5-58. DCSM_Z1_OTP Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-n		Value after reset or the default value

5.9.5.1 Z1OTP_LINKPOINTER1 Register (Offset = 0h) [Reset = FFFFFFFFh]

Z1OTP_LINKPOINTER1 is shown in Figure 5-51 and described in Table 5-59.

Return to the Summary Table.

Zone 1 Link Pointer1

Figure 5-51. Z1OTP_LINKPOINTER1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1OTP_LINKPOINTER1																															
R-FFFFFFFFh																															

Table 5-59. Z1OTP_LINKPOINTER1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1OTP_LINKPOINTER1	R	FFFFFFFFh	Zone1 Link Pointer 1 location in USER OTP. Note: [1] ECC comparison is disabled for this location [2] When this value is loaded into DCSM, if the bits[31:14] !=0, device will remain in BLOCKED state. Before shipping parts to customers, TI would change the value of these bits to 0s. Reset type: N/A

5.9.5.2 Z1OTP_LINKPOINTER2 Register (Offset = 2h) [Reset = FFFFFFFFh]

Z1OTP_LINKPOINTER2 is shown in Figure 5-52 and described in Table 5-60.

Return to the Summary Table.

Zone 1 Link Pointer2

Figure 5-52. Z1OTP_LINKPOINTER2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1OTP_LINKPOINTER2																															
R-FFFFFFFFh																															

Table 5-60. Z1OTP_LINKPOINTER2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1OTP_LINKPOINTER2	R	FFFFFFFFh	Zone1 Link Pointer 2 location in USER OTP. Note: [1] ECC comparison is disabled for this location [2] When this value is loaded into DCSM, if the bits[31:14] !=0, device will remain in BLOCKED state. Before shipping parts to customers, TI would change the value of these bits to 0s. Reset type: N/A

5.9.5.3 Z1OTP_LINKPOINTER3 Register (Offset = 4h) [Reset = FFFFFFFFh]

Z1OTP_LINKPOINTER3 is shown in Figure 5-53 and described in Table 5-61.

Return to the Summary Table.

Zone 1 Link Pointer3

Figure 5-53. Z1OTP_LINKPOINTER3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1OTP_LINKPOINTER3																															
R-FFFFFFFFh																															

Table 5-61. Z1OTP_LINKPOINTER3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1OTP_LINKPOINTER3	R	FFFFFFFFh	Zone1 Link Pointer 3 location in USER OTP. Note: [1] ECC comparison is disabled for this location [2] When this value is loaded into DCSM, if the bits[31:14] !=0, device will remain in BLOCKED state. Before shipping parts to customers, TI would change the value of these bits to 0s. Reset type: N/A

5.9.5.4 Z1OTP_JLM_ENABLE Register (Offset = 6h) [Reset = FFFFFFFFh]

Z1OTP_JLM_ENABLE is shown in Figure 5-54 and described in Table 5-62.

Return to the Summary Table.

Zone 1 JTAGLOCK Enable Register

Figure 5-54. Z1OTP_JLM_ENABLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1OTP_JLM_ENABLE																															
R-FFFFFFFFh																															

Table 5-62. Z1OTP_JLM_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1OTP_JLM_ENABLE	R	FFFFFFFFh	Zone1 JLM_ENABLE register location in USER OTP. Note: When this value is loaded into Z1_JLM_ENABLE, if the value is 32-bit all-1s, the JTAGLOCK will be enabled. Before shipping parts to customers, TI will program the default value to 0xFFFF_000F, which will disable the JTAGLOCK feature. Users should program 0xFFFF_0000 to enable the JTAGLOCK feature. Reset type: N/A

5.9.5.5 Z1OTP_GPREG1 Register (Offset = 8h) [Reset = FFFFFFFFh]

Z1OTP_GPREG1 is shown in Figure 5-55 and described in Table 5-63.

Return to the Summary Table.

Zone 1 General Purpose Register 1

Figure 5-55. Z1OTP_GPREG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1OTP_GPREG1																															
R-FFFFFFFFh																															

Table 5-63. Z1OTP_GPREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1OTP_GPREG1	R	FFFFFFFFh	Zone1 General Purpose register location in USER OTP. Reset type: N/A

5.9.5.6 Z1OTP_GPREG2 Register (Offset = Ah) [Reset = FFFFFFFFh]

Z1OTP_GPREG2 is shown in Figure 5-56 and described in Table 5-64.

Return to the Summary Table.

Zone 1 General Purpose Register 2

Figure 5-56. Z1OTP_GPREG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1OTP_GPREG2																															
R-FFFFFFFFh																															

Table 5-64. Z1OTP_GPREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1OTP_GPREG2	R	FFFFFFFFh	Zone1 General Purpose register location in USER OTP. Reset type: N/A

5.9.5.7 Z1OTP_GPREG3 Register (Offset = Ch) [Reset = FFFFFFFFh]

Z1OTP_GPREG3 is shown in Figure 5-57 and described in Table 5-65.

Return to the Summary Table.

Zone 1 General Purpose Register 3

Figure 5-57. Z1OTP_GPREG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1OTP_GPREG3																															
R-FFFFFFFFh																															

Table 5-65. Z1OTP_GPREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1OTP_GPREG3	R	FFFFFFFFh	Zone1 General Purpose register location in USER OTP. Reset type: N/A

5.9.5.8 Z1OTP_GPREG4 Register (Offset = Eh) [Reset = FFFFFFFFh]

Z1OTP_GPREG4 is shown in Figure 5-58 and described in Table 5-66.

Return to the Summary Table.

Zone 1 General Purpose Register 4

Figure 5-58. Z1OTP_GPREG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1OTP_GPREG4																															
R-FFFFFFFFh																															

Table 5-66. Z1OTP_GPREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1OTP_GPREG4	R	FFFFFFFFh	Zone1 General Purpose register location in USER OTP. Reset type: N/A

5.9.5.9 Z1OTP_PSWDLOCK Register (Offset = 10h) [Reset = FFFFFFFFh]

Z1OTP_PSWDLOCK is shown in Figure 5-59 and described in Table 5-67.

Return to the Summary Table.

Secure Password Lock

Figure 5-59. Z1OTP_PSWDLOCK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z1OTP_PSWDLOCK																															
R-FFFFFFFFh																															

Table 5-67. Z1OTP_PSWDLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z1OTP_PSWDLOCK	R	FFFFFFFFh	Zone1 password lock location in USER OTP. Note: When this value is loaded into DCSM, if the value is 32-bit all-1s, CSMPSWD will remain locked. Before shipping parts to customers, TI would change the value of this location in such a way that the ECC field remains all-1s and also LSB 4-bits remain 4'b1111. Reset type: N/A

5.9.5.10 Z1OTP_JTAGPSWDH0 Register (Offset = 14h) [Reset = FFFFFFFFh]

Z1OTP_JTAGPSWDH0 is shown in Figure 5-60 and described in Table 5-68.

Return to the Summary Table.

JTAG Lock Permanent Password 0

Figure 5-60. Z1OTP_JTAGPSWDH0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGPSWDH0																															
R-FFFFFFFFh																															

Table 5-68. Z1OTP_JTAGPSWDH0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGPSWDH0	R	FFFFFFFFh	JTAG Lock Password High 0 (bits 95:64) location in USER Z1 OTP. This value is dummy loaded into the non-memory-mapped JTAGPSWD register, bits 95:64. TI must program a default value into this location, leaving the ECC bits all 1's. Reset type: N/A

5.9.5.11 Z1OTP_JTAGPSWDH1 Register (Offset = 16h) [Reset = FFFFFFFFh]

Z1OTP_JTAGPSWDH1 is shown in Figure 5-61 and described in Table 5-69.

Return to the Summary Table.

JTAG Lock Permanent Password 1

Figure 5-61. Z1OTP_JTAGPSWDH1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAGPSWDH1																															
R-FFFFFFFFh																															

Table 5-69. Z1OTP_JTAGPSWDH1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	JTAGPSWDH1	R	FFFFFFFFh	JTAG Lock Password High 1 (bits 127:96) location in USER Z1 OTP. This value is dummy loaded into the non-memory-mapped JTAGPSWD register, bits 127:96. Reset type: N/A

5.9.5.12 Z1OTP_CMACKEY0 Register (Offset = 18h) [Reset = FFFFFFFFh]

Z1OTP_CMACKEY0 is shown in Figure 5-62 and described in Table 5-70.

Return to the Summary Table.

Secure Boot CMAC Key 0

Figure 5-62. Z1OTP_CMACKEY0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMACKEY0																															
R-FFFFFFFFh																															

Table 5-70. Z1OTP_CMACKEY0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMACKEY0	R	FFFFFFFFh	Secure Boot CMAC Key 0 (bits 31:0) location in User Z1 OTP. This value is dummy loaded into the CMACKEY0 register. Reset type: N/A

5.9.5.13 Z1OTP_CMACKEY1 Register (Offset = 1Ah) [Reset = FFFFFFFFh]

Z1OTP_CMACKEY1 is shown in Figure 5-63 and described in Table 5-71.

Return to the Summary Table.

Secure Boot CMAC Key 1

Figure 5-63. Z1OTP_CMACKEY1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMACKEY1																															
R-FFFFFFFFh																															

Table 5-71. Z1OTP_CMACKEY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMACKEY1	R	FFFFFFFFh	Secure Boot CMAC Key 1 (bits 63:32) location in User Z1 OTP. This value is dummy loaded into the CMACKEY1 register. Reset type: N/A

5.9.5.14 Z1OTP_CMACKEY2 Register (Offset = 1Ch) [Reset = FFFFFFFFh]

Z1OTP_CMACKEY2 is shown in Figure 5-64 and described in Table 5-72.

Return to the Summary Table.

Secure Boot CMAC Key 2

Figure 5-64. Z1OTP_CMACKEY2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMACKEY2																															
R-FFFFFFFFh																															

Table 5-72. Z1OTP_CMACKEY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMACKEY2	R	FFFFFFFFh	Secure Boot CMAC Key 2 (bits 95:64) location in User Z1 OTP. This value is dummy loaded into the CMACKEY2 register. Reset type: N/A

5.9.5.15 Z1OTP_CMACKEY3 Register (Offset = 1Eh) [Reset = FFFFFFFFh]

Z1OTP_CMACKEY3 is shown in Figure 5-65 and described in Table 5-73.

Return to the Summary Table.

Secure Boot CMAC Key 3

Figure 5-65. Z1OTP_CMACKEY3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMACKEY3																															
R-FFFFFFFFh																															

Table 5-73. Z1OTP_CMACKEY3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMACKEY3	R	FFFFFFFFh	Secure Boot CMAC Key 3 (bits 127:96) location in User Z1 OTP. This value is dummy loaded into the CMACKEY3 register. Reset type: N/A

5.9.6 DCSM_Z2_OTP Registers

Table 5-74 lists the memory-mapped registers for the DCSM_Z2_OTP registers. All register offset addresses not listed in Table 5-74 should be considered as reserved locations and the register contents should not be modified.

Table 5-74. DCSM_Z2_OTP Registers

Offset	Acronym	Register Name	Write Protection
0h	Z2OTP_LINKPOINTER1	Zone 2 Link Pointer1	
2h	Z2OTP_LINKPOINTER2	Zone 2 Link Pointer2	
4h	Z2OTP_LINKPOINTER3	Zone 2 Link Pointer3	
8h	Z2OTP_GPREG1	Zone 2 General Purpose Register 1	
Ah	Z2OTP_GPREG2	Zone 2 General Purpose Register 2	
Ch	Z2OTP_GPREG3	Zone 2 General Purpose Register 3	
Eh	Z2OTP_GPREG4	Zone 2 General Purpose Register 4	
10h	Z2OTP_PSWDLOCK	Secure Password Lock	

Complex bit access types are encoded to fit into small table cells. Table 5-75 shows the codes that are used for access types in this section.

Table 5-75. DCSM_Z2_OTP Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-n		Value after reset or the default value

5.9.6.1 Z2OTP_LINKPOINTER1 Register (Offset = 0h) [Reset = FFFFFFFFh]

Z2OTP_LINKPOINTER1 is shown in Figure 5-66 and described in Table 5-76.

Return to the Summary Table.

Zone 2 Link Pointer1

Figure 5-66. Z2OTP_LINKPOINTER1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2OTP_LINKPOINTER1																															
R-FFFFFFFFh																															

Table 5-76. Z2OTP_LINKPOINTER1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2OTP_LINKPOINTER1	R	FFFFFFFFh	Zone2 Link Pointer 1 location in USER OTP. Note: [1] ECC comparison is disabled for this location [2] When this value is loaded into DCSM, if the bits[31:14] !=0, device will remain in BLOCKED state. Before shipping parts to customers, TI would change the value of these bits to 0s. Reset type: N/A

5.9.6.2 Z2OTP_LINKPOINTER2 Register (Offset = 2h) [Reset = FFFFFFFFh]

Z2OTP_LINKPOINTER2 is shown in Figure 5-67 and described in Table 5-77.

Return to the Summary Table.

Zone 2 Link Pointer2

Figure 5-67. Z2OTP_LINKPOINTER2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2OTP_LINKPOINTER2																															
R-FFFFFFFFh																															

Table 5-77. Z2OTP_LINKPOINTER2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2OTP_LINKPOINTER2	R	FFFFFFFFh	Zone2 Link Pointer 2 location in USER OTP. Note: [1] ECC comparison is disabled for this location [2] When this value is loaded into DCSM, if the bits[31:14] !=0, device will remain in BLOCKED state. Before shipping parts to customers, TI would change the value of these bits to 0s. Reset type: N/A

5.9.6.3 Z2OTP_LINKPOINTER3 Register (Offset = 4h) [Reset = FFFFFFFFh]

Z2OTP_LINKPOINTER3 is shown in Figure 5-68 and described in Table 5-78.

Return to the Summary Table.

Zone 2 Link Pointer3

Figure 5-68. Z2OTP_LINKPOINTER3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2OTP_LINKPOINTER3																															
R-FFFFFFFFh																															

Table 5-78. Z2OTP_LINKPOINTER3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2OTP_LINKPOINTER3	R	FFFFFFFFh	Zone2 Link Pointer 3 location in USER OTP. Note: [1] ECC comparison is disabled for this location [2] When this value is loaded into DCSM, if the bits[31:14] !=0, device will remain in BLOCKED state. Before shipping parts to customers, TI would change the value of these bits to 0s. Reset type: N/A

5.9.6.4 Z2OTP_GPREG1 Register (Offset = 8h) [Reset = FFFFFFFFh]

Z2OTP_GPREG1 is shown in Figure 5-69 and described in Table 5-79.

Return to the Summary Table.

Zone 2 General Purpose Register 1

Figure 5-69. Z2OTP_GPREG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2OTP_GPREG1																															
R-FFFFFFFFh																															

Table 5-79. Z2OTP_GPREG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2OTP_GPREG1	R	FFFFFFFFh	Zone2 General Purpose register location in USER OTP. Reset type: N/A

5.9.6.5 Z2OTP_GPREG2 Register (Offset = Ah) [Reset = FFFFFFFFh]

Z2OTP_GPREG2 is shown in Figure 5-70 and described in Table 5-80.

Return to the Summary Table.

Zone 2 General Purpose Register 2

Figure 5-70. Z2OTP_GPREG2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2OTP_GPREG2																															
R-FFFFFFFFh																															

Table 5-80. Z2OTP_GPREG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2OTP_GPREG2	R	FFFFFFFFh	Zone2 General Purpose register location in USER OTP. Reset type: N/A

5.9.6.6 Z2OTP_GPREG3 Register (Offset = Ch) [Reset = FFFFFFFFh]

Z2OTP_GPREG3 is shown in Figure 5-71 and described in Table 5-81.

Return to the Summary Table.

Zone 2 General Purpose Register 3

Figure 5-71. Z2OTP_GPREG3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2OTP_GPREG3																															
R-FFFFFFFFh																															

Table 5-81. Z2OTP_GPREG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2OTP_GPREG3	R	FFFFFFFFh	Zone2 General Purpose register location in USER OTP. Reset type: N/A

5.9.6.7 Z2OTP_GPREG4 Register (Offset = Eh) [Reset = FFFFFFFFh]

Z2OTP_GPREG4 is shown in Figure 5-72 and described in Table 5-82.

Return to the Summary Table.

Zone 2 General Purpose Register 4

Figure 5-72. Z2OTP_GPREG4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2OTP_GPREG4																															
R-FFFFFFFFh																															

Table 5-82. Z2OTP_GPREG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2OTP_GPREG4	R	FFFFFFFFh	Zone2 General Purpose register location in USER OTP. Reset type: N/A

5.9.6.8 Z2OTP_PSWDLOCK Register (Offset = 10h) [Reset = FFFFFFFFh]

Z2OTP_PSWDLOCK is shown in Figure 5-73 and described in Table 5-83.

Return to the Summary Table.

Secure Password Lock

Figure 5-73. Z2OTP_PSWDLOCK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Z2OTP_PSWDLOCK																															
R-FFFFFFFFh																															

Table 5-83. Z2OTP_PSWDLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Z2OTP_PSWDLOCK	R	FFFFFFFFh	Zone2 password lock location in USER OTP. Note: When this value is loaded into DCSM, if the value is 32-bit all-1s, CSMPSWD will remain locked. Before shipping parts to customers, TI would change the value of this location in such a way that the ECC field remains all-1s and also LSB 4-bits remain 4'b1111. Reset type: N/A

Chapter 6

Flash Module

This chapter describes the Flash module.

6.1 Introduction to Flash and OTP Memory

Flash is an electrically erasable/programmable nonvolatile memory that can be programmed and erased many times to ease code development. Flash memory can be used primarily as a program memory for the core, and secondarily as static data memory.

This section describes the proper sequence to configure the wait states and operating mode of Flash. This section also includes information on one-time programmable (OTP) Flash, how to improve Flash performance by enabling the Flash prefetch/cache mode, and the SECDED safety feature.

6.1.1 FLASH Related Collateral

Foundational Materials

- C28x Academy - FLASH
- *Calculating Useful Lifetimes of Embedded Processors Application Report*
- Embedded Flash Memory (Video)

Getting Started Materials

- *Serial Flash Programming of C2000 Microcontrollers Application Report*
- [FAQ] FAQ for Flash ECC usage in C2000 devices - Includes ECC test mode, Linker ECC options:
- [FAQ] FAQ on Flash API usage for C2000 devices
- [FAQ] Flash - How to modify an application from RAM configuration to Flash configuration?
- [FAQ] How can we improve the Flash tool performance?
- [FAQ] TI C2000 Device Programming Tools and Services

6.1.2 Features

Features of Flash memory include:

- One Flash bank (Bank0) (refer to the device data sheet for the size of the Flash bank)
- Configurable Flash programming option and ECC
- User-programmable OTP locations (in user-configurable DCSM OTP, also called USER OTP) for configuring security, OTP boot-mode and boot-mode select pins (if the user is unable to use the factory-default boot-mode select pins)
- Enhanced performance when using the code prefetch mechanism and data cache.
- Configurable wait states to give the best performance for a given SYSCLK frequency
- Safety Features:
  - SECDED - single-error correction and double-error detection is supported
  - Address bits are included in ECC
  - Test mode to check the health of ECC logic
- Flash Wrapper for controlling the Flash bank
  - Integrated Flash program/erase state machine
  - Simple Flash API algorithms
  - Fast erase and program times (refer to the device data sheet for details)
- Dual Code Security Module (DCSM) to prevent unauthorized access to the Flash (refer to the *Dual Code Security Module (DCSM)* chapter for details)

6.1.3 Flash Tools

Texas Instruments provides the following tools for Flash:

- Code Composer Studio™ (CCS) IDE - development environment with integrated Flash plugin. TI recommends performing a debug reset and restart before programming the code into Flash using CCS.
- Flash API Library - set of software peripheral functions to erase/program Flash.
- UniFlash - standalone tool to erase/program/verify the Flash content through JTAG. No CCS usage is required.

Note

Users must check and install available updates for the CCS On-Chip Flash Plugin and UniFlash tools.

6.1.4 Default Flash Configuration

The following are Flash module configuration settings at power-up:

- Flash Bank and Pump are powered up on device reset.
- ECC is enabled
- Wait-states are set to the maximum (0xF)
- Code-prefetch mechanism and data cache are disabled

During the boot process, the boot ROM performs a dummy read of the Code Security Module (CSM) password locations in the OTP. This read is performed to unlock a new device that has no password stored in the device, so that Flash programming or loading of code into CSM-protected SRAM can be performed. On devices with a password, this read has no effect and the device remains locked.

To achieve optimum system performance, user application software must initialize wait-states using the FRDCNTL register, and configure cache/prefetch features using the FRD_INTF_CTRL register. Software that configures Flash settings like wait-states, cache/prefetch features, and so on, must be executed only from RAM memory, **not** from Flash memory.

Note

Before initializing wait-states, turn off the pre-fetch and data caching in the FRD_INTF_CTRL register.

6.2 Flash Bank, OTP, and Pump

This device includes one Flash bank. In addition, there is one-time programmable Flash memory (OTP) in the Flash bank. Flash and OTP are uniformly mapped in both program and data memory space.

There are two OTP regions. The first OTP region, TI-OTP, contains manufacturing information, trims, Flash operation settings, and other device data. TI-OTP can be read by the user application, but TI-OTP cannot be programmed or erased. The second OTP region, USER-OTP, is primarily used for programming device security (DCSM module) settings. USER-OTP can be programmed only once and cannot be erased afterwards. For information on the memory-map, Flash bank sizes, TI-OTP, USER-OTP, and corresponding ECC locations, refer to the device data sheet.

The Figure 5-2 in the *Dual Code Security Module (DCSM)* chapter shows the user-programmable OTP locations in CPU1 USER-OTP. For more information on the functionality of these fields, refer to the *ROM Code and Peripheral Booting* chapter and the *Dual Code Security Module (DCSM)* chapter.

6.3 Flash Wrapper

The CPU interfaces with the Flash wrapper, which interfaces with the Flash bank and the pump (see Figure 6-1). The Flash wrapper has the following primary features:

- Provides a simple interface for software to program or erase the Flash memory.
- Provides an interface for the CPU to read Flash data, including data caching and prefetch features.
- Performs ECC error checking and correction, and generates interrupts when an error is detected.
- Provides the capability to prevent unwanted bank program or erase operations.

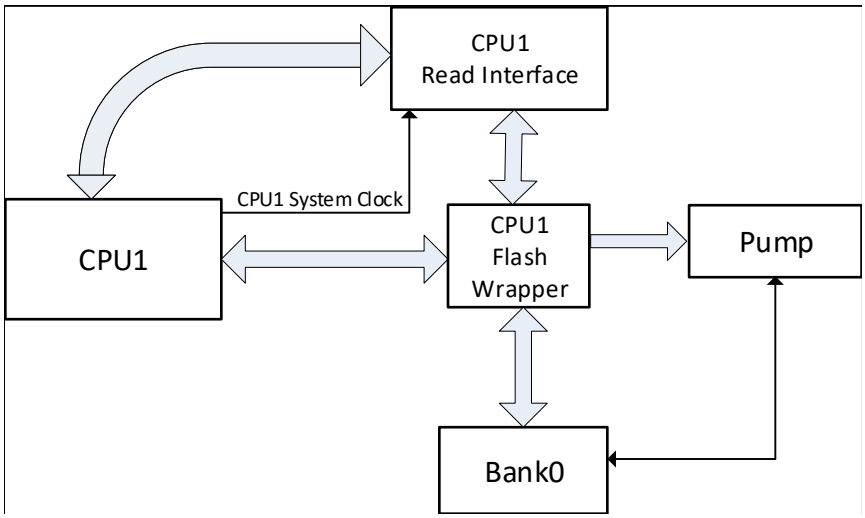


Figure 6-1. Flash Interface Block Diagram

6.4 Flash and OTP Memory Performance

Flash read or instruction fetch accesses can be classified either as a Flash access (access to an address location in Flash), or an OTP access (access to an address location in OTP).

When the CPU performs an access to a Flash memory address, data is returned after (RWAIT + 1) SYSCLK cycles.

RWAIT defines the number of random access wait states, and is configured using the RWAIT field in the FRDCNTL register. At reset, RWAIT defaults to a worst-case wait state count (15), and therefore must be initialized to the appropriate number of wait states to improve performance, based on the CPU clock frequency and the access time of the Flash. The Flash supports zero-wait accesses when RWAIT is set to zero, when the CPU clock frequency is low enough to accommodate the Flash access time.

For a given system clock frequency, configure RWAIT using the following formula:

For C28x Flash Bank: RWAIT = ceiling[(SYSCLK/FCLK) - 1]

where SYSCLK is the system operating frequency for CPU1, and FCLK is the clock frequency for Flash.

FCLK must be ≤ $FCLK_{max}$, the allowed maximum Flash clock frequency at RWAIT = 0.

If RWAIT results in a fractional value when calculated using the above formula, round up RWAIT to the nearest integer.

6.5 Flash Read Interface

This section provides details about the data read modes to access Flash bank/OTP and the configuration registers that control the read interface. In addition to a standard read mode, the Flash wrapper has a built-in prefetch and cache mechanism to allow increased clock speeds and CPU throughput wherever applicable.

6.5.1 C28x-Flash Read Interface

6.5.1.1 Standard Read Mode

Standard read mode is the default Flash read mode after reset. In this mode, the code prefetch mechanism and data cache are disabled. When standard read mode is active, every read access to Flash is decoded by the Flash wrapper to fetch the data from the addressed location, and the data is returned after RWAIT + 1 cycles (except User OTP).

Flash data buffers associated with the prefetch mechanism and data cache are bypassed in standard read mode; therefore, every access to the Flash/OTP is used by the CPU immediately, and every access creates a unique Flash bank access.

Standard read mode is the recommended mode for lower system frequency operation, where RWAIT can be set to zero to provide single-cycle access operation. The Flash wrapper can operate at higher frequencies using standard read mode, at the expense of adding wait states. At higher system frequencies, it is recommended to enable the data cache and prefetch mechanisms to improve performance. Refer to the device data sheet to determine the maximum Flash frequency allowed in standard read mode (that is, maximum Flash clock frequency with RWAIT = 0, $FCLK_{MAX}$).

6.5.1.2 Prefetch Mode

Flash memory is typically used to store application code. During code execution, instructions are fetched from contiguous memory addresses, except when a discontinuity occurs. Usually, the portion of the code that resides in contiguous address locations makes up the majority of the application code, and is referred to as linear code. To improve the performance of linear code execution, a Flash prefetch mechanism has been implemented. Figure 6-2 illustrates how this mode functions.

The prefetch mechanism does a look-ahead prefetch on linear address increments, starting from the address of the last instruction fetch. The Flash prefetch mechanism is disabled by default. To enable prefetch mode, set the PREFETCH_EN bit in the FRD_INTF_CTRL register, or call the Flash_enablePrefetch() driverlib function.

Each instruction fetch from the Flash or OTP reads out 128 bits. The starting address of the access from Flash is automatically aligned to a 128-bit boundary, such that the instruction location is within the 128 bits to be fetched. When Flash prefetch mode is enabled, the 128 bits read from the instruction fetch are stored in a 128-bit wide by 2-level deep instruction prefetch buffer. The contents of this prefetch buffer are then sent to the CPU for processing as required.

Up to four 32-bit or eight 16-bit instructions can reside within a single 128-bit access. The majority of C28x instructions are 16 bits, so for every 128-bit instruction fetch from the Flash bank there are up to eight instructions in the prefetch buffer ready to process through the CPU. During the time to process these instructions, the Flash prefetch mechanism automatically initiates another access to the Flash bank to prefetch the next 128 bits. The Flash prefetch mechanism works in the background to keep the instruction prefetch buffers as full as possible. Using this technique, the overall efficiency of sequential code execution from Flash or OTP is improved significantly.

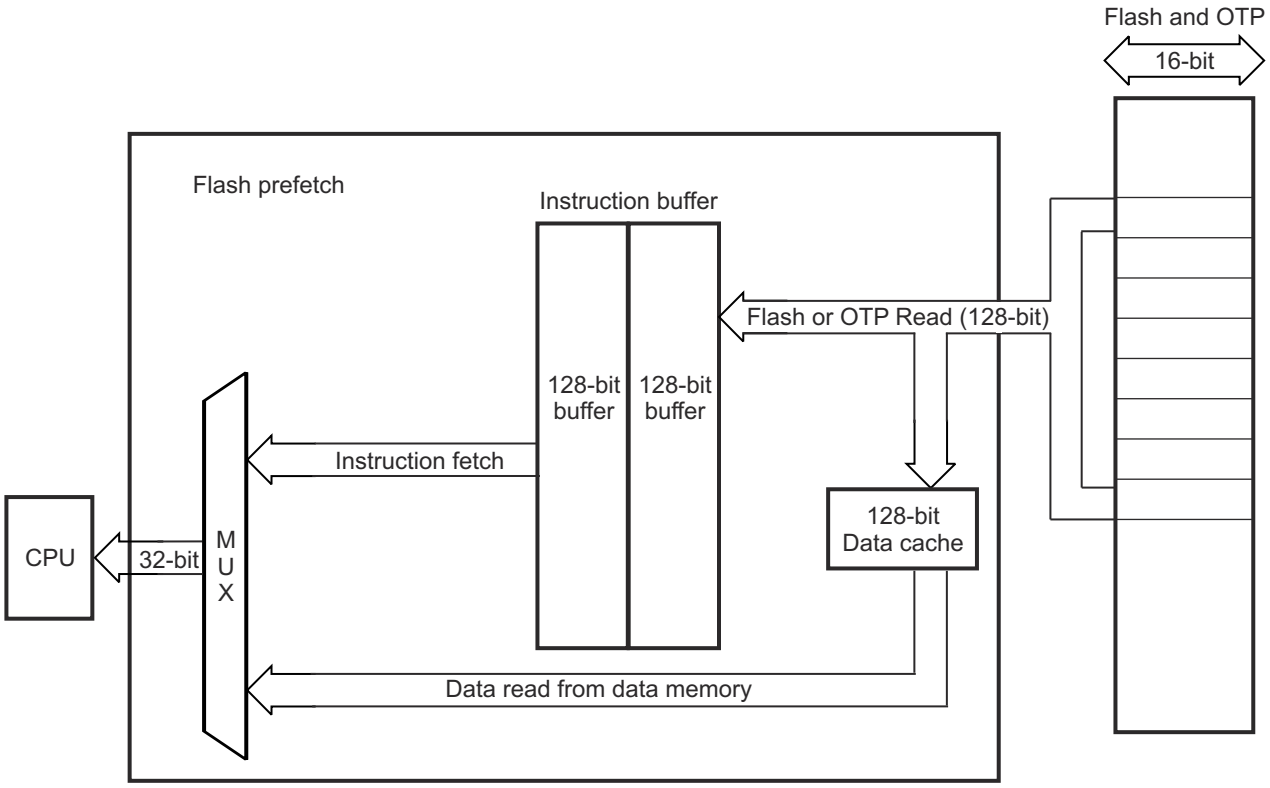


Figure 6-2. Flash Prefetch Mode

The Flash prefetch is aborted only when there is a code discontinuity caused by executing an instruction such as a branch, function call, or loop. When this occurs, the prefetch mechanism is aborted, and the contents of the prefetch buffer are flushed. There are two possible scenarios when this occurs:

1. If the destination address is within the Flash or OTP, the prefetch aborts and then resumes at the destination address.
2. If the destination address is outside of the Flash and OTP, the prefetch is aborted, and begins again only when the code branches back into the Flash or OTP. The Flash prefetch mechanism only applies to instruction fetches from program space. Data reads from data memory and from program memory do not utilize the prefetch mechanism and thus bypass the prefetch buffer. For example, instructions such as MAC, DMAC, and PREAD read a data value from program memory. When such a read happens, the prefetch buffer is bypassed, but the buffer is not flushed. If an instruction prefetch is already in progress when a data read operation is initiated, then the data read is stalled until the prefetch completes.

Note that the prefetch mechanism gets bypassed when RWAIT is configured as zero.

6.5.1.3 Data Cache

In addition to the prefetch mechanism, a data cache of 128-bits wide has been implemented to improve data space read performance. This data cache is separate from the instruction prefetch buffer, and is used for data reads only. Whenever a data read access is performed by the CPU to a Flash bank address, if the data located at that address is not presently loaded into the data cache, then the Flash wrapper reads 128 bits of data from the Flash bank and stores the data in the data cache. This data is eventually sent to the CPU for processing. The starting address of the Flash bank access is automatically aligned to a 128-bit boundary, such that the requested address location is within the 128 bits to be read from the bank.

The data cache is disabled by default at reset. To enable the data cache, set the DATA_CACHE_EN bit in the FRD_INTF_CTRL register, or call the Flash_enableCache() driverlib function. Note that the data cache gets bypassed when RWAIT is set to zero.

Note

The data cache does not get updated on a debugger access, or when a read to the ECC memory-mapped region is performed.

6.5.1.4 Flash Read Operation

There are a few important points to keep in mind when using Flash or OTP memory:

- Reads of USER OTP locations are hardwired for wait states. The RWAIT bits have no effect on these locations.
- CPU writes to Flash or OTP memory addresses are ignored, and complete within a single cycle. Flash memory can only be modified by issuing program or erase commands, using the Flash API.
- When a security zone is in the locked state, and the respective password lock bits are not all ones, then:
  - Data reads to Zx-CSMPSWD return zero;
  - Program space reads to Zx-CSMPSWD return zero; and
  - Program fetches to Zx-CSMPSWD return zero.
- When the Code Security Module (CSM) is secured, reads to Flash or OTP memory addresses from outside the secure zone take the same number of cycles as a normal access. However, the read operation returns zero.
- The arbitration scheme in the Flash wrapper prioritizes CPU accesses in the fixed priority order of data space read (highest priority), program space read, and program fetches/program prefetches (lowest priority).
- When Flash state machine is activated for erase or program operations, the contents of the prefetch buffer and data cache are automatically flushed.

Note

ECC checks are performed on data read from Flash before the data is stored in the prefetch buffer or data cache. Once data has entered the cache or buffer, there are no further ECC checks performed.

6.6 Flash Erase and Program

Flash memory can be programmed either by using the CCS Flash plug-in or by using the UniFlash application. If these methods are not feasible in an application, the Flash API can be used. The Flash memory can be programmed, erased, and verified only by using the Flash API library. These functions are written, compiled and validated by Texas Instruments. The Flash module contains a Flash state machine (FSM) to perform program and erase operations.

The recommended flow for programming Flash is:
Erase → Program → Verify

6.6.1 Erase

When the target flash is erased, the Flash reads as all 1s. This state is called 'blank.' The erase function must be executed before programming. Only bank erase is available for this device. However, to erase at the sector level, protection masks can be used with the bank erase function. The user cannot skip erase on sectors that read as 'blank' because these sectors can require additional erasing due to marginally erased bits columns. The erase function erases the data and the ECC together.

Note

Providing the correct sector mask for the bank erase command is important. If the mask is mistakenly chosen to erase an inaccessible sector (belongs to another security zone), the bank erase command continues attempting to erase the sector endlessly and the FSM never exits (since erase does not succeed). To avoid such a situation, the user must take care to provide the correct mask. However, given that there is a chance of choosing an incorrect mask, TI suggests to initialize the maximum allowed erase pulses to zero after the maximum number of pulses are issued by the FSM for the bank erase operation. This makes sure that the FSM ends the bank erase command after trying to erase the inaccessible sector up to the maximum allowed erase pulses.

The Example_EraseBanks() function in the C2000Ware's Flash API usage example depicts the implementation of this sequence (content of the while loop waiting for the FSM to complete the bank erase command). Users must use this code as-is irrespective of whether or not security is used by the application to also make sure that the FSM exits from bank erase operations in case of an erase-failure.

6.6.2 Program

The Flash wrapper provides a command to program the Flash and User OTP. This command is also used to program ECC check bits.

Note

The main array Flash programming must be aligned to 64-bit address boundaries, and each 64-bit word can only be programmed once per write/erase cycle.

The DCSM OTP programming must be aligned to 128-bit address boundaries and each 128-bit word can only be programmed once. The exceptions are:

- The DCSM Zx-LINKPOINTER1 and Zx-LINKPOINTER2 values in the DCSM OTP can be programmed together and can be programmed one bit at a time as required by the DCSM operation.
- The DCSM Zx-LINKPOINTER3 values in the DCSM OTP can be programmed one bit at a time as required by the DCSM operation.

To avoid exceeding data retention capability limits, do not perform more than 4 program operations on the same Flash word line before performing an erase operation. Each Flash word line consists of sixteen 128-bit words (256 bytes). This limit is especially important to observe when writing to one-time-programmable/non-erasable Flash regions, such as the User OTP.

6.6.3 Verify

After programming, the user should perform verification using API function Fapi_doVerify(). This function verifies the Flash contents against supplied data.

Application software typically performs a CRC check of the Flash memory contents during power-up and at regular intervals during runtime (as needed). Apart from this, ECC logic, when enabled (enabled by default), catches single-bit errors, double-bit errors, and address errors whenever the CPU reads/fetches from a Flash address.

6.7 Error Correction Code (ECC) Protection

There are two ECC blocks (ECC64_H and ECC64_L) inside the Flash Read Interface. These ECC blocks correct single-bit Flash read errors, and can detect uncorrectable errors of two or more bits. The ECC blocks are also capable of detecting address errors. The ECC blocks operate using eight user-calculated ECC check bits associated with each 64-bit wide data word and the corresponding 128-bit memory-aligned address. Users must program these ECC check bits into ECC memory space, along with Flash data during the Flash programming operation. Refer to the device data sheet for the Flash/OTP ECC memory-map.

The ECC bits for a given Flash address and 64-bit data word can be calculated using the Flash API; however, TI recommends using the AutoEccGeneration option available in the Flash Plugin or API to auto-calculated and program ECC bits. The Flash API uses hardware ECC logic in the device to generate the ECC data for the given Flash data. The Flash Plugin, the Flash programming tool integrated with the Code Composer Studio™ IDE, uses the Flash API to generate and program ECC data.

Figure 6-3 illustrates the ECC logic inputs and outputs.

During an instruction fetch or a data read operation, the 19 most-significant address bits (the three least-significant bits of address are not considered), together with the 64-bit data/8-bit ECC read-out of Flash banks/ECC memory-map area, pass through the ECC logic, and the eight check bits are produced in ECC block. These eight calculated ECC check bits are then XORed with the stored check bits (user programmed check bits) associated with the address and the read data. The 8-bit output is decoded inside the ECC block to determine one of three conditions:

- No error occurred
- A correctable error (single bit data error) occurred

- A non-correctable error (double bit data error or address error) occurred

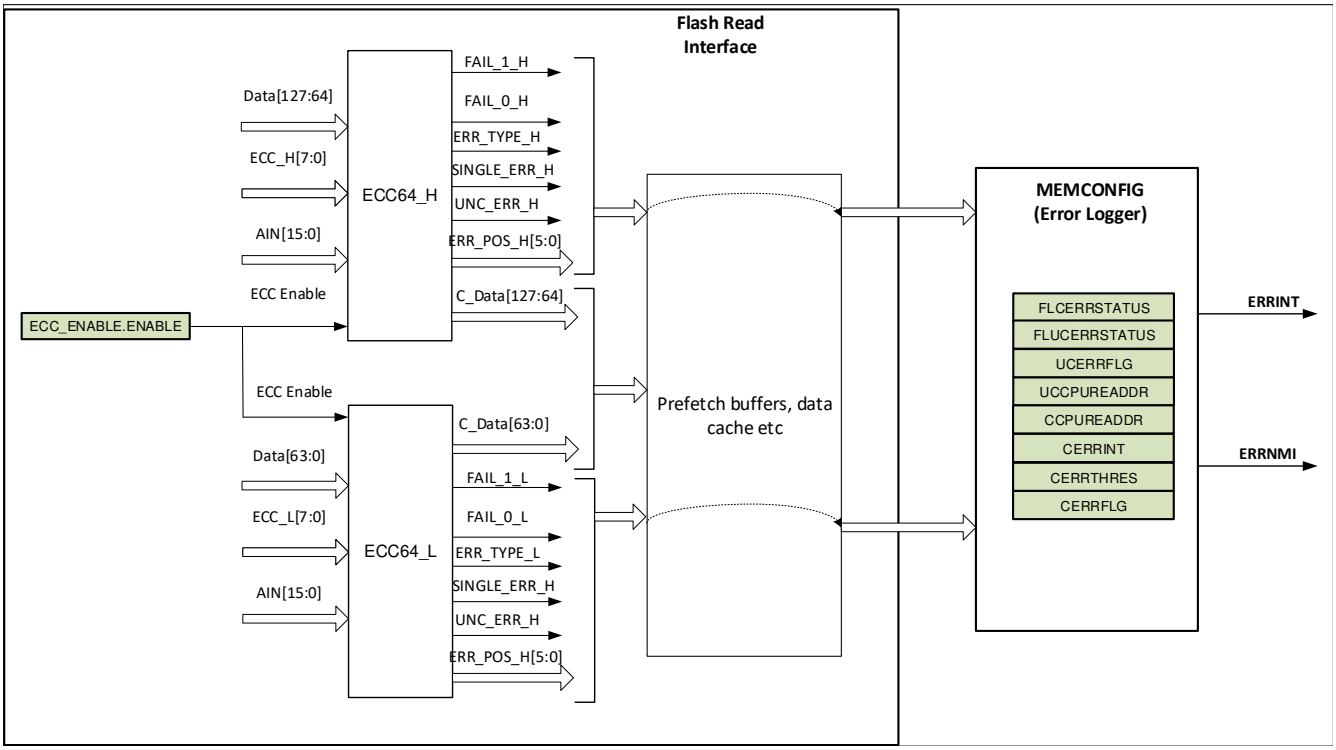


Figure 6-3. ECC Logic Inputs and Outputs

A single-bit error in the address field is considered to be a non-correctable error.

Note

Since ECC is calculated for an entire 64-bit data word, a non 64-bit read such as a byte read or a half-word read still forces the entire 64-bit data word to be read and calculated, even though only the byte or half-word is actually used by the CPU.

The ECC feature is enabled by default at reset, and can be enabled or disabled by writing to the ECC_ENABLE register. ECC logic is automatically bypassed when the 64 data bits and associated ECC bits fetched from the bank are either all ones or all zeros.

6.7.1 Single-Bit Data Error

This section provides information for both single-bit data errors and single-bit ECC check bit errors. If there is a single bit flip (0 to 1 or 1 to 0) in Flash data or in ECC data, then the error is considered as a single-bit data error. The SECDED module detects and corrects single-bit errors, if any, in the 64-bit Flash data or eight ECC check bits read from the Flash/ECC memory-map before the read data is provided to the CPU.

When SECDED finds and corrects single bit data errors, the following information is logged in the ECC registers if the ECC feature is enabled:

- Address where the error occurred: if the single-bit error occurs in the lower 64 bits of a 128-bit memory-aligned Flash data word, the address of the lower 64-bit word is captured in the SINGLE_ERR_ADDR_LOW register. If the single-bit error occurs in the upper 64 bits of the 128-bit data word, then the address of the upper 64-bit word is captured in the SINGLE_ERR_ADDR_HIGH register.
- Whether the error occurred in data bits or ECC bits: the ERR_TYPE_L and ERR_TYPE_H bit fields in the ERR_POS register indicate whether the error occurred in data bits or ECC bits of the lower 64 bits, or the upper 64 bits respectively, of a 128-bit memory-aligned Flash data word.
- Bit position at which the error occurred: the ERR_POS_L and ERR_POS_H bit fields in the ERR_POS register indicate the bit position of the error in the lower 64 bits/lower 8-bit ECC, or the upper 64 bits/upper 8-bit ECC respectively, of a 128-bit memory-aligned Flash data word.
- Whether the corrected value is 0 (FAIL_0_L, FAIL_0_H flags in ERR_STATUS register).
- Whether the corrected value is 1 (FAIL_1_L, FAIL_1_H flags in ERR_STATUS register).
- A single bit error counter that increments on every single bit error occurrence (ERR_CNT register) until a user-configurable threshold (see ERR_THRESHOLD) is met.
- A flag that gets set when one or more single-bit errors occurs after ERR_CNT equals ERR_THRESHOLD (SINGLE_ERR_INT_FLG flag in the ERR_INTFLG register).

When the ERR_CNT value equals ERR_THRESHOLD+1, and a single bit error occurs, the Flash module sets the SINGLE_ERR_INT flag and generates an interrupt signal. To enable propagation of the generated interrupt pulse to the CPU, the user application must enable the FLASH_CORRECTABLE_ERROR channel in the C28 Peripheral Interrupt Expansion module (PIE). The interrupt signal remains high until the application clears the SINGLE_ERR_INTFLG flag by writing to the SINGLE_ERR_INTCLR bit in the ERR_INTCLR register. The Flash module cannot generate any further FLASH_CORRECTABLE_ERROR interrupt signals to the PIE/CPU until SINGLE_ERR_INTFLG is cleared, as this is an edge-based interrupt.

When multiple single-bit errors have been detected by ECC logic, the contents of the Flash ECC registers reflect the most recent ECC error. When multiple single-bit errors have been detected, both FAIL_0_L and FAIL_1_L (or FAIL_0_H and FAIL_1_H) can be set, indicating that single-bit fail0/fail1 occurred in different 64-bit aligned addresses.

Although ECC is calculated on 64-bit basis, a read of any address location within a 128-bit aligned Flash data word causes the single-bit error flag to get set, if there is a single-bit error in both or in either the lower 64 or upper 64 bits (or corresponding ECC check bits) of that 128-bit data word.

6.7.2 Uncorrectable Error

Uncorrectable errors include address errors and double-bit errors in data or ECC. When the ECC logic finds uncorrectable errors, the following information is logged in ECC registers if the ECC feature is enabled:

- Address where the error occurred: if the uncorrectable error occurs in the lower 64 bits of a 128-bit memory-aligned Flash data word, the lower 64-bit memory-aligned address is captured in the UNC_ERR_ADDR_LOW register. If the uncorrectable error occurs in the upper 64 bits of a 128-bit memory-aligned Flash data word, the upper 64-bit memory-aligned address is captured in the UNC_ERR_ADDR_HIGH register.
- A flag is set indicating that an uncorrectable error occurred – the UNC_ERR_L and UNC_ERR_H flags in the ERR_STATUS register indicate the uncorrectable error occurrence in the lower 64 bits/lower 8-bit ECC, or the upper 64 bits/upper 8-bit ECC, respectively, of a 128-bit memory-aligned Flash data word.
- A flag is set indicating that an uncorrectable error interrupt is generated (UNC_ERR_INTFLG in ERR_INTFLG register).

When an uncorrectable error occurs, the Flash module sets the UNC_ERR_INTFLG bit and generates an uncorrectable error interrupt. This uncorrectable error interrupt generates a non-maskable interrupt (NMI), if enabled, in the CPU. If an uncorrectable error interrupt flag is not cleared by writing to the UNC_ERR_INTCLR bit in the ERR_INTCLR register, the Flash module cannot generate new uncorrectable interrupt signals, as this is an edge-based interrupt.

Although ECC is calculated on 64-bit basis, a read of any address location within a 128-bit aligned Flash word causes the uncorrectable error flag to get set, and an uncorrectable error interrupt/NMI to occur, when there is a uncorrectable error in both or in either the lower 64 bits or upper 64 bits (or corresponding ECC check bits) of that 128-bit data word.

6.7.3 ECC Logic Self Test

A mechanism has been added to enable self-testing of the ECC logic for additional diagnostic coverage. To use this mechanism, configure the ECC_TEST_EN field in the FECC_CTRL register. A value of 01 in ECC_TEST_EN injects a single-bit error into the redundant ECC logic upon a Flash read access, and a value of 11 injects a double-bit error upon a Flash read access. This causes an output comparison failure. In this mode, the diagnostic outputs of each of the high and low comparators (DIAG_H and DIAG_L) are captured in the FLUCERRSTATUS Memconfig register.

Note

When ECC self-test is enabled and CPU issues a read access to the Flash, ECC errors are captured in the data cache and prefetch buffers. TI recommends that application software disables caching while performing diagnostic checks.

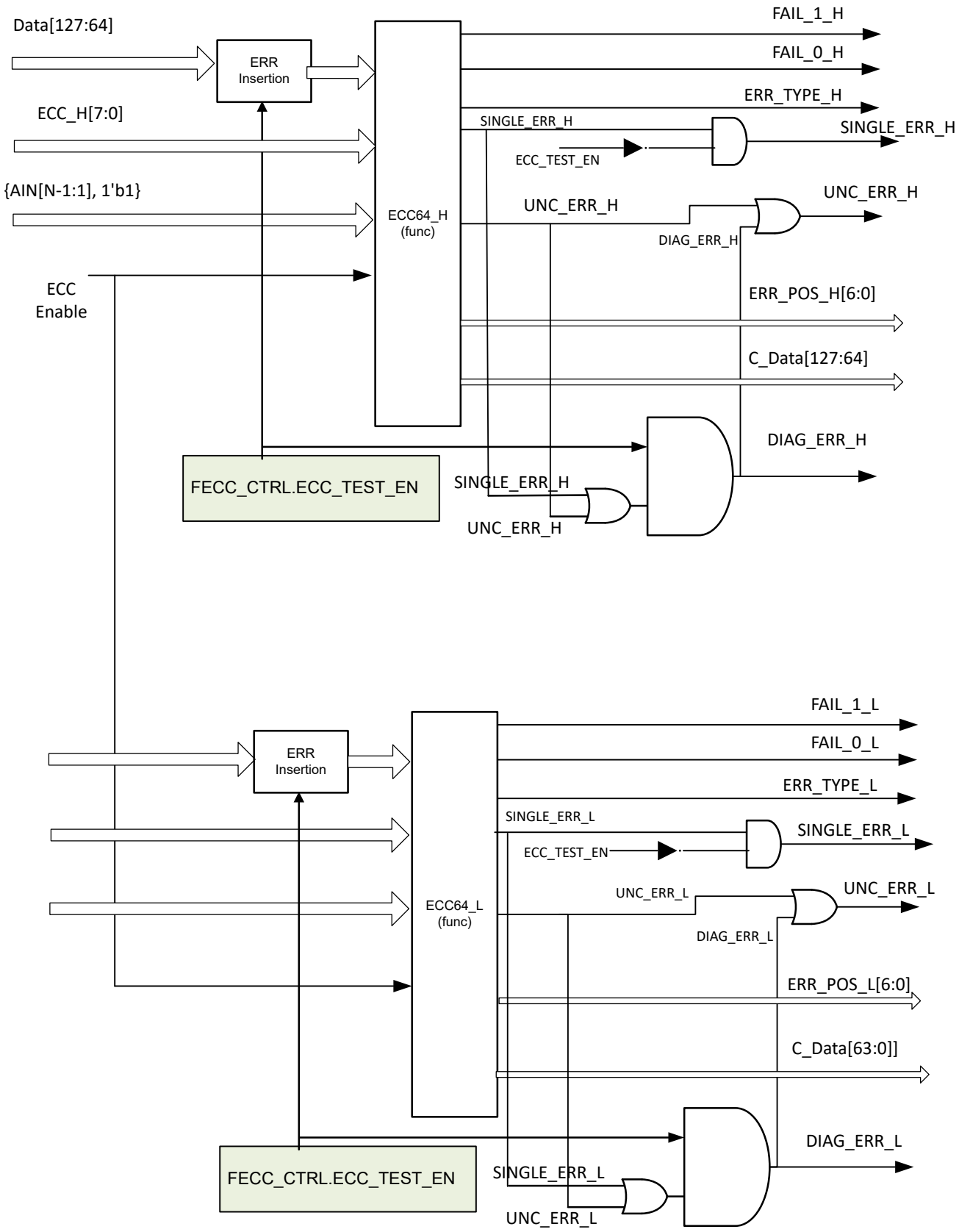


Figure 6-4. Testing ECC Logic

6.8 Reserved Locations Within Flash and OTP

When allocating code and data to Flash and OTP memory, keep the following reserved locations in mind:

- The entire OTP has reserved user-configurable locations for security and boot process. For more details on the functionality of these fields, refer to the *ROM Code and Peripheral Booting* chapter and the *Dual Code Security Module (DCSM)* chapter.
- Refer to the *ROM Code and Peripheral Booting* chapter for reserved locations in Flash for real-time operating system usage and a boot-to-Flash entry point. A boot-to-Flash entry point is reserved for an entry-into-Flash branch instruction. When the boot-to-Flash boot option is used, the boot ROM jumps to this address in Flash. If you program a branch instruction here, that redirects code execution to the entry point of the application.

6.9 Migrating an Application from RAM to Flash

To migrate an existing application that is configured to run from RAM to a Flash-based linker configuration, follow these steps:

1. Replace the RAM linker command file with a Flash linker command file. For examples of Flash-based linker command files, see the device_support\<device>\common\cmd directory.
2. When modifying the Flash-based linker command file, be sure to map any initialized sections to Flash memory regions.
3. Make sure the boot mode pins are configured for Flash boot. This tells the boot ROM to redirect execution to the application programmed into Flash memory after boot code execution is complete. For more information on boot mode configuration, see *Detailed Description* > *Device Boot Modes* in the device data sheet.
4. When the device is configured for Flash boot, the boot ROM redirects execution to the Flash entry point location (defined as BEGIN in TI-provided Flash linker command files) at the end of boot code execution. Make sure there is a branch instruction at the Flash entry point to your code initialization (for example, _c_int00) function. In the C2000Ware examples, the entry point code is specified in the codestartbranch.asm file.
5. To achieve best performance for Flash execution, configure the Flash wait states as per the device operating clock frequency, as specified in the device data sheet. In addition, enable prefetch mode and data cache mode. Calling the Flash_initModule() driverlib function achieves these steps. Note that code that initializes the Flash module must execute from a RAM location. This is accomplished by assigning the Flash initialization function to the .TI.ramfunc section. In the linker command file, map this section to Flash for load, and RAM for execution. The example cmd files provided in the C2000Ware show how to do this correctly.
6. For any functions that require 0- or 1-wait state performance, be sure to map to RAM for execution in the linker command file, similar to the Flash initialization function. The .TI.ramfunc section in the TI-provided Flash linker command files accomplishes this purpose.
7. Align all code and data sections to 128-bit address boundaries when mapping to Flash memory, using the ALIGN directive in the linker command file.
8. For EABI executable formats, define all uninitialized sections mapped to RAM as NOINIT sections (using the directive "type=NOINIT") in the linker command file.
9. Be sure to program ECC bits correctly for the Flash application image. Keep the AutoEccGeneration option enabled in the Code Composer Studio Flash Plugin or UniFlash GUI.

6.10 Procedure to Change the Flash Control Registers

During Flash configuration, no accesses to the Flash or OTP can be in progress. This includes instructions still in the CPU pipeline, data reads, and instruction prefetch operations. To be sure that no access takes place during the configuration change, follow the procedure shown below for any code that modifies the Flash control registers.

1. Start executing application code from RAM/Flash/OTP.
2. Branch to or call the Flash configuration code (that writes to Flash control registers) in RAM. This is required to properly flush the CPU pipeline before the configuration change. The function that changes the Flash configuration cannot execute from the Flash or OTP and must reside in RAM.
3. Execute the Flash configuration code (located in RAM) that writes to Flash control registers like FRDCNTL, FRD_INTF_CTRL, and so on.
4. At the end of the Flash configuration code execution, wait eight cycles to let the write instructions propagate through the CPU pipeline. This must be done before the return-from-function call is made.
5. Return to the calling function that resides in RAM or Flash/OTP and continue execution.

6.11 Software

6.11.1 FLASH Examples

NOTE: These examples are located in the C2000ware Software Development Kit installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/flash

Cloud access to these examples is available at the following link: dev.ti.com C2000ware Examples.

6.11.1.1 Flash Programming with AutoECC, DataAndECC, DataOnly and EccOnly

||END||

This example demonstrates how to program Flash using API's following options
1. AutoEcc generation

External Connections
- None.

Watch Variables
- None.

6.12 FLASH Registers

This Section describes the FLASH Registers.

6.12.1 FLASH Base Address Table

Table 6-1. FLASH Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
Flash0CtrlRegs	FLASH_CTRL_REGS	FLASH0CTRL_BASE	0x0005_F800	-	YES
Flash0EccRegs	FLASH_ECC_REGS	FLASH0ECC_BASE	0x0005_FB00	-	YES

6.12.2 FLASH_CTRL_REGS Registers

Table 6-2 lists the memory-mapped registers for the FLASH_CTRL_REGS registers. All register offset addresses not listed in Table 6-2 should be considered as reserved locations and the register contents should not be modified.

Table 6-2. FLASH_CTRL_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	FRDCNTL	Flash Read Control Register	EALLOW
2h	FLPROT	Flash program/erase protect register	EALLOW
6h	FRD_INTF_CTRL	Flash Read Interface Control Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 6-3 shows the codes that are used for access types in this section.

Table 6-3. FLASH_CTRL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

6.12.2.1 FRDCNTL Register (Offset = 0h) [Reset = 0F000F00h]

FRDCNTL is shown in Figure 6-5 and described in Table 6-4.

Return to the Summary Table.

Flash Read Control Register

Figure 6-5. FRDCNTL Register

31	30	29	28	27	26	25	24
RESERVED				RESERVED			
R-0h				R/W-Fh			
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				RWAIT			
R-0h				R/W-Fh			
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 6-4. FRDCNTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	RESERVED	R/W	Fh	Reserved
23-12	RESERVED	R	0h	Reserved
11-8	RWAIT	R/W	Fh	Random read waitstate These bits indicate how many waitstates are added to a flash read/fetch access. The RWAIT value can be set anywhere from 0 to 0xF. For a flash access, data is returned in RWAIT+1 SYSCLK cycles. Note: The required wait states for each SYSCLK frequency can be found in the device data manual. Reset type: SYSRSn
7-0	RESERVED	R	0h	Reserved

6.12.2.2 FLPROT Register (Offset = 2h) [Reset = 00000000h]

FLPROT is shown in Figure 6-6 and described in Table 6-5.

Return to the Summary Table.

Flash program/erase protect register

Figure 6-6. FLPROT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							FLWEPROT
R-0h							R/W-0h

Table 6-5. FLPROT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	FLWEPROT	R/W	0h	Flash program/erase protect bit. 0 : Program erase operation allowed subject to security settings. 1 : Program erase operation blocked in hardware. Reset type: SYSRSn

6.12.2.3 FRD_INTF_CTRL Register (Offset = 6h) [Reset = 00000000h]

FRD_INTF_CTRL is shown in Figure 6-7 and described in Table 6-6.

Return to the Summary Table.

Flash Read Interface Control Register

Figure 6-7. FRD_INTF_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						DATA_CACHE_EN	PREFETCH_EN
R-0h						R/W-0h	R/W-0h

Table 6-6. FRD_INTF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-2	RESERVED	R	0h	Reserved
1	DATA_CACHE_EN	R/W	0h	Data cache enable. 0 A value of 0 disables the data cache. 1 A value of 1 enables the data cache. Reset type: SYSRSn
0	PREFETCH_EN	R/W	0h	Prefetch enable. 0 A value of 0 disables prefetch mechanism. 1 A value of 1 enables pre-fetch mechanism. Reset type: SYSRSn

6.12.3 FLASH_ECC_REGS Registers

Table 6-7 lists the memory-mapped registers for the FLASH_ECC_REGS registers. All register offset addresses not listed in Table 6-7 should be considered as reserved locations and the register contents should not be modified.

Table 6-7. FLASH_ECC_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	ECC_ENABLE	ECC Enable	EALLOW
2h	FECC_CTRL	ECC Control	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 6-8 shows the codes that are used for access types in this section.

Table 6-8. FLASH_ECC_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

6.12.3.1 ECC_ENABLE Register (Offset = 0h) [Reset = 0000000Ah]

ECC_ENABLE is shown in Figure 6-8 and described in Table 6-9.

Return to the Summary Table.

ECC Enable

Figure 6-8. ECC_ENABLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												ENABLE			
R-0h												R/W-Ah			

Table 6-9. ECC_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	ENABLE	R/W	Ah	ECC enable. A value of 0xA would enable ECC. Any other value would disable ECC. Reset type: SYSRSn

6.12.3.2 FECC_CTRL Register (Offset = 2h) [Reset = 00000000h]

FECC_CTRL is shown in Figure 6-9 and described in Table 6-10.

Return to the Summary Table.

ECC Control

Figure 6-9. FECC_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						ECC_TEST_EN	
R-0h						R/W-0h	

Table 6-10. FECC_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-2	RESERVED	R	0h	Reserved
1-0	ECC_TEST_EN	R/W	0h	ECC test mode enable. 00 ECC test mode disabled 01 ECC test mode enabled, one of the 64 data bits is flipped and fed to the redundant ECC logic (on both ECC logic low and ECC logic high blocks). 11 ECC test mode enabled, Two of the 64 data bits are flipped and fed to the redundant ECC logic (on both ECC logic low and ECC logic high blocks). 10 Reserved Reset type: SYSRSn

Chapter 7

Dual-Clock Comparator (DCC)

This chapter describes the Dual-Clock Comparator (DCC) module.

7.1 Introduction

The dual-clock comparator module is used for evaluating and monitoring the clock input based on a second clock, which can be a more accurate and reliable version. This instrumentation is used to detect faults in clock source or clock structures, thereby enhancing the system's safety metrics.

7.1.1 Features

The main features of each of the DCC modules are:

- Allows the application to make sure that a fixed ratio is maintained between frequencies of two clock signals.
- Supports the definition of a programmable tolerance window in terms of the number of reference clock cycles.
- Supports continuous monitoring without requiring application intervention.
- Supports a single-sequence mode for spot measurements.
- Allows the selection of a clock source for each of the counters, resulting in several specific use cases.

7.1.2 Block Diagram

Figure 7-1 shows how the DCC connects to the rest of the system. Figure 7-2 shows the main concept of the DCC module.

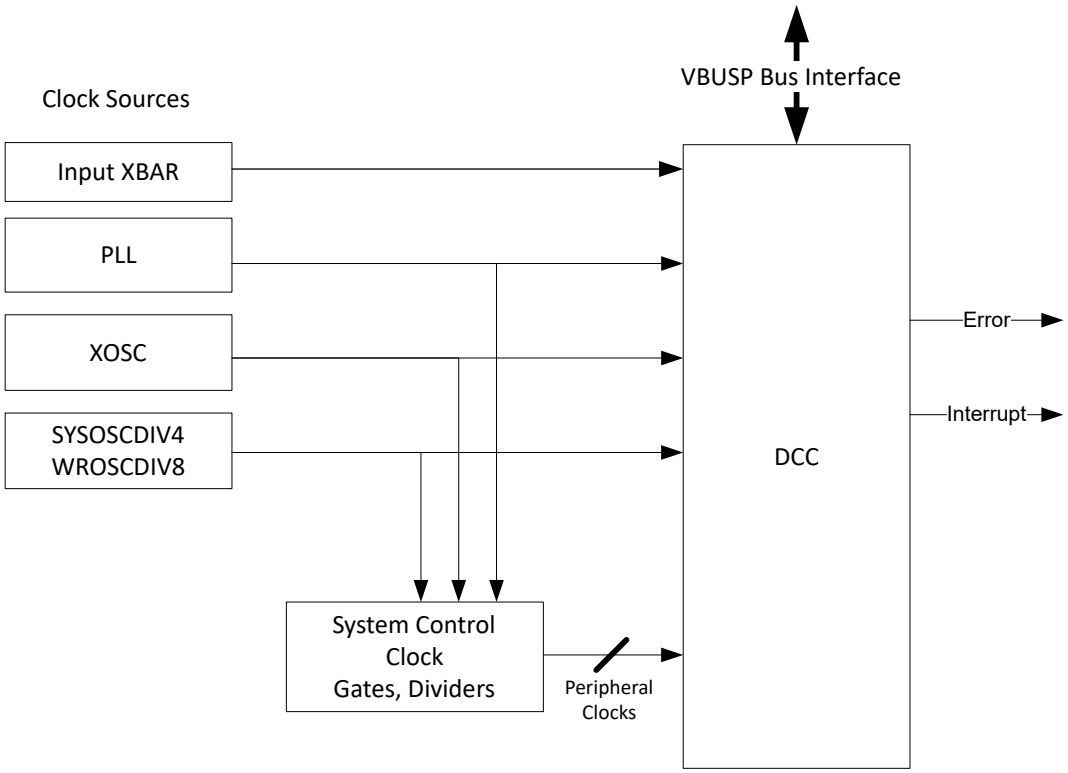


Figure 7-1. DCC Module Overview

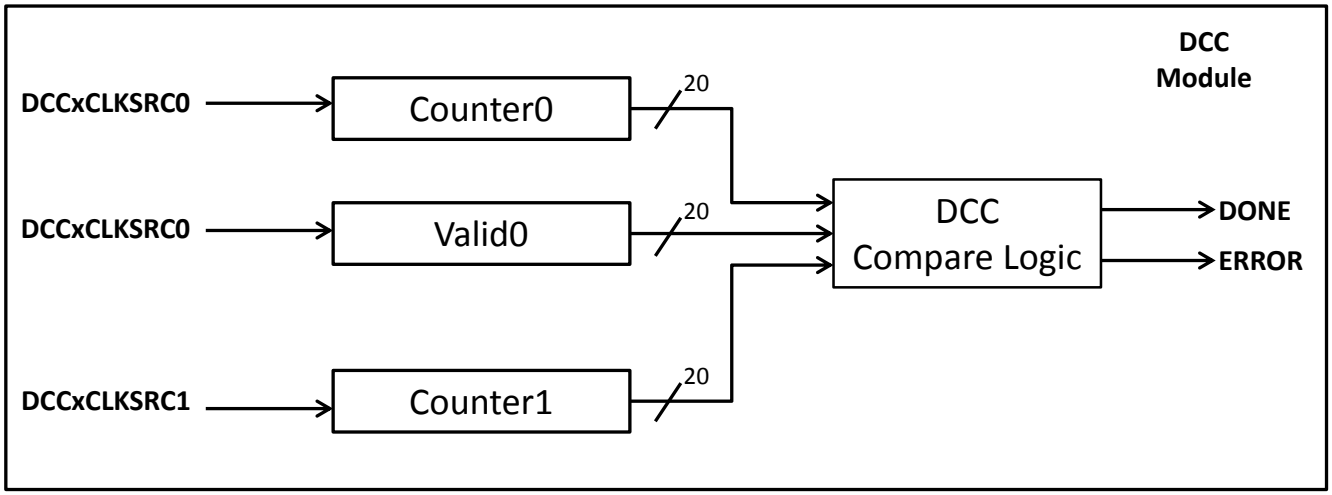


Figure 7-2. DCC Operation

7.2 Module Operation

As shown in Figure 7-2, DCC contains three counters – Counter0, Valid0 and Counter1. Initially, all counters are loaded with the user-defined, pre-load value. Counter0 and Counter1 start decrementing once the DCC is enabled at rates determined by the frequencies of Clock0 and Clock1, respectively. When Counter0 equals 0 (expires), the Valid0 counter decrements at a rate determined by Clock0. If Counter1 decrements to 0 in the valid window, then no error is generated and Clock1 is considered to be good within allowable tolerance as configured by the user.

7.2.1 Configuring DCC Counters

Counter0 and Counter1 are configured based on the ratio between the frequencies of Clock0 and Clock1 (Fclk1 × Counter0 = Fclk0×Counter1). The Valid0 counter provides tolerance and is configured based on the error in DCC. Since Clock0 and Clock1 are asynchronous, the start and stop of the counters do not occur synchronously. Hence, while configuring the counters, two different sources of errors must be accounted for:

- DCC Errors due to the asynchronous timing of Clock0 and Clock1: this depends on the frequency of Clock0 and Clock1:
  - If Fclk1 > Fclk0, then Async. Error (in Clock0 cycles) = 2 + 2 × (Fsysclk/Fclk0)
  - If Fclk1 < Fclk0, then Async. Error (in Clock0 cycles) = 2 × (Fclk0/Fclk1) + 2 × (Fsysclk/Fclk0)
  - If Fclk1 is unknown, then Async. Error (in Clock0 cycles) = 2 + 2 × (Fsysclk/Fclk0)
- Digitization Error = 8 Clock0 cycles

DCC Error (in Clock0 Cycles) = Async. Error + Digitization Error

DCC error shows up as a frequency error for clock under measurement. This error is DCC induced and does not represent error in frequency of clock under measurement. The application needs to take this into consideration while configuring the counters, and determine a desirable tolerance for DCC error that defines the window of measurement. To illustrate:

Window (in Clock0 Cycles) = (DCC Error)/(0.01 × Tolerance)

For example, if DCC Error is 10 and the tolerance desired is ±0.1%, then:

Window (in Clock0 Cycles) = 10/(0.01 × 0.1) = 10000

Based on above formula for Window, if the desired tolerance is low, then the counter values are large and increase the window of measurement. This means that counter values for a tolerance of 0.1% are larger than that of 0.2%. So, based on the application defined tolerance, define the window of measurement in terms of Clock0 cycles.

The clock under measurement can have an allowed frequency error. If this error is expected, then the error can also be accounted while configuring counters. For example, if measuring SYSOSCDIV4 frequency using an external crystal as a reference clock, the allowable tolerance of SYSOSCDIV4 (for example, ±1%) can be accounted for and factored into the counter configuration. The formula is:

Frequency Error Allowed (in Clock0 Cycles) = Window × (Allowable Frequency Tolerance (in %) / 100)

Total Error (in Clock0 Cycles) = DCC Error + Frequency Error Allowed

The following equations are used to configure counter values:

Counter0 (DCCCNTSEED0) = Window - Total Error

Valid0 (DCCVALIDSEED0) = 2 × Total Error

Counter1 (DCCCNTSEED1) = Window × (Fclk1/Fclk0)

Note

Counter1 is a 20-bit counter, so the maximum possible value cannot exceed 1048575. If the value does exceed, then increase the desired Tolerance for DCC error, so that Window of measurement is lowered. The following formula can be used to compute minimum tolerance possible:

Tolerance (%) = (100 × DCC Error × (Fclk1/Fclk0)) / 1048575

7.2.2 Single-Shot Measurement Mode

The DCC module can be programmed to count down one time by enabling the single-shot mode. In this mode, the DCC stops operating when the down counter0 and the valid counter0 reach 0.

At the end of one sequence of counting down in this single-shot mode, the DCC gets disabled automatically, which prevents further counting. This mode is typically used for spot-checking the frequency of a signal.

Example-1: Validating PLLRAWCLK frequency

A practical example of the usage is to validate the PLL output clock frequency using the XTAL as the reference clock. Assume XTAL is 10MHz, PLL output frequency is 100MHz, SYSCLK is 100MHz, allowable Frequency Tolerance is 0.1%, and DCC Tolerance required is 0.1%. The measurement sequence proceeds as follows:

- Set Clock0 source for Counter0 and Valid0 as XTAL, and Clock1 source for Counter1 as PLL output clock.
- Based on the equations defined in Section 7.2.1, calculated seed values for Counters can be Counter0 = 29940; Valid0 = 120; Counter1 = 300000
- Once the DCC is enabled, the counters Counter0 and Counter1 both start counting down from the seed values.
- When Counter0 reaches zero, Counter0 automatically triggers the Valid0 counter.
- When Valid0 reaches zero and Counter1 is not zero, an ERROR status flag is set and a "DCC error" is sent to the . Counter1 is frozen so that the counter stops counting down any further. The application can enable an interrupt to be generated from the whenever this DCC error is indicated.
- The application then needs to clear the ERROR status flag and restart the DCC module so that the module is ready for the next spot measurement.

If there is no error generated at the end of the sequence, then the DONE status flag is set and a DONE interrupt is generated. The application must clear the DONE flag before restarting the DCC.

Error Conditions:

An error condition is generated by any one of the following:

1. Counter1 counts down to 0 before Counter0 reaches 0. This means that Clock1 is faster than expected, or Clock0 is slower than expected. This error includes the case when Clock0 is stuck at 1 or 0.
2. Counter1 does not reach 0 even when Counter0 and Valid0 have both reached 0. This means that Clock1 is slower than expected. This error includes the case when Clock1 is stuck at 1 or 0.

Any error freezes the counters from counting. An application can then read out the counter values to help determine what caused the error.

7.2.3 Continuous Monitoring Mode

In this mode, the DCC is used by the application to make sure that two clock signals maintain the correct frequency ratio. Suppose the application wants to make sure that the PLL output signal always maintains a fixed frequency relationship with the XTAL:

- In this case, the application can use the XTAL as the Clock0 signal (for Counter0 and Valid0) and the PLL output as the Clock1 (for Counter1).
- The seed values of Counter0, Valid0 and Counter1 are selected based on the equations defined in Section 7.2.1 such that if the actual frequencies of Clock0 and Clock1 are equal to the expected frequencies, then the Counter1 reaches zero during the count down of the Valid0 counter.
- If the Counter1 reaches zero during the count down of the Valid0 counter, then all the counters (Counter0, Valid0, Counter1) are reloaded with the initial seed values.
- This sequence of counting down and checking then continues as long as there is no error, or until the DCC module is disabled.
- The counters must get reloaded if the application resets and restarts the DCC module.

Error Conditions:

An error condition is generated by one of the following:

1. Counter1 counts down to 0 before Counter0 reaches 0. This means that Clock1 is faster than expected or Clock0 is slower than expected. This condition includes the case when Clock0 is stuck at 1 or 0.
2. Counter1 does not reach 0 even when Counter0 and Valid0 have both reached 0. This means that Clock1 is slower than expected. This condition includes the case when Clock1 is stuck at 1 or 0.

Any error freezes the counters from counting. An application can then read out the counter values to help determine what caused the error.

7.2.4 Error Conditions

While operating in continuous mode, the counters get reloaded with the seed values and continue counting down under the following conditions:

- The module is reset or restarted by the application, OR
- Counter0, Valid 0, and Counter1 all reach 0 without any error.

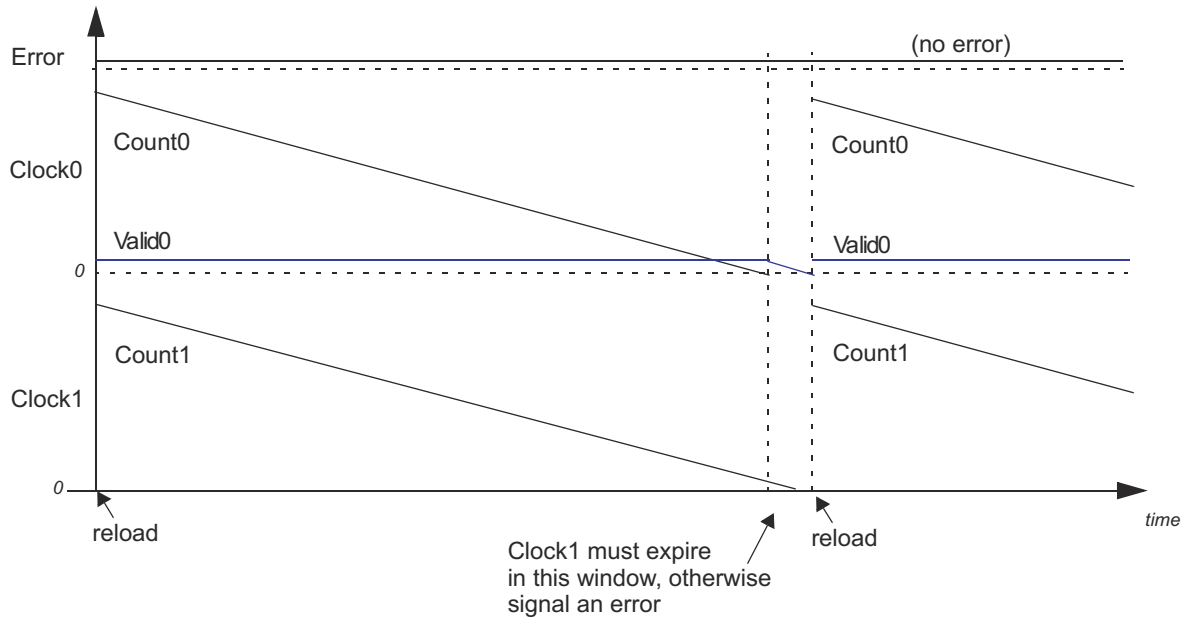


Figure 7-3. Counter Relationship

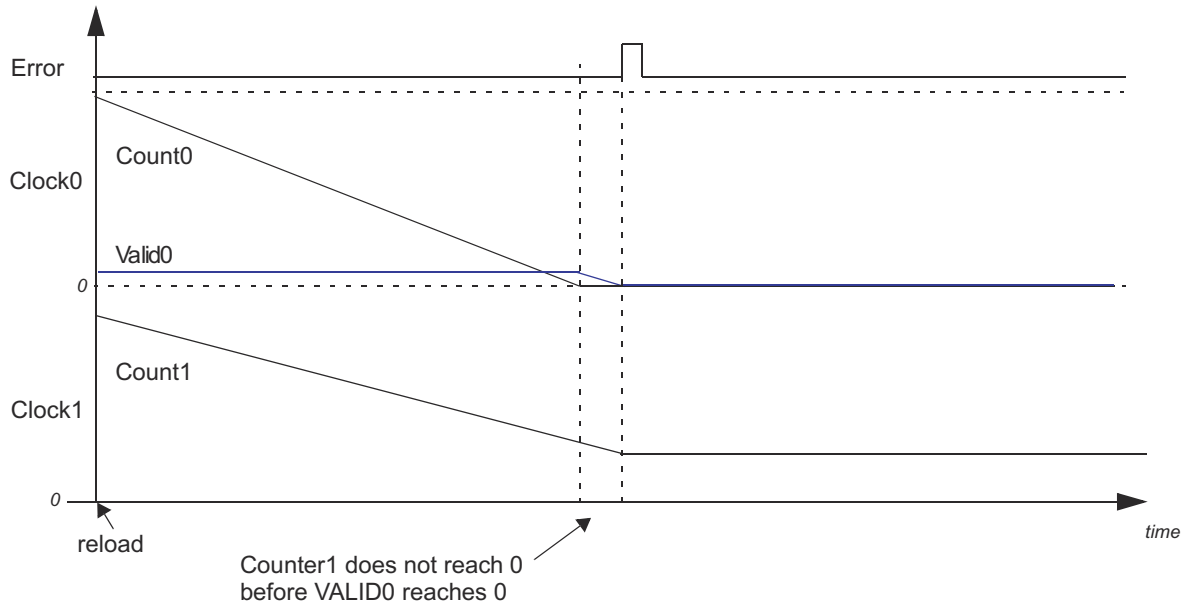


Figure 7-4. Clock1 Slower Than Clock0 - Results in an Error and Stops Counting

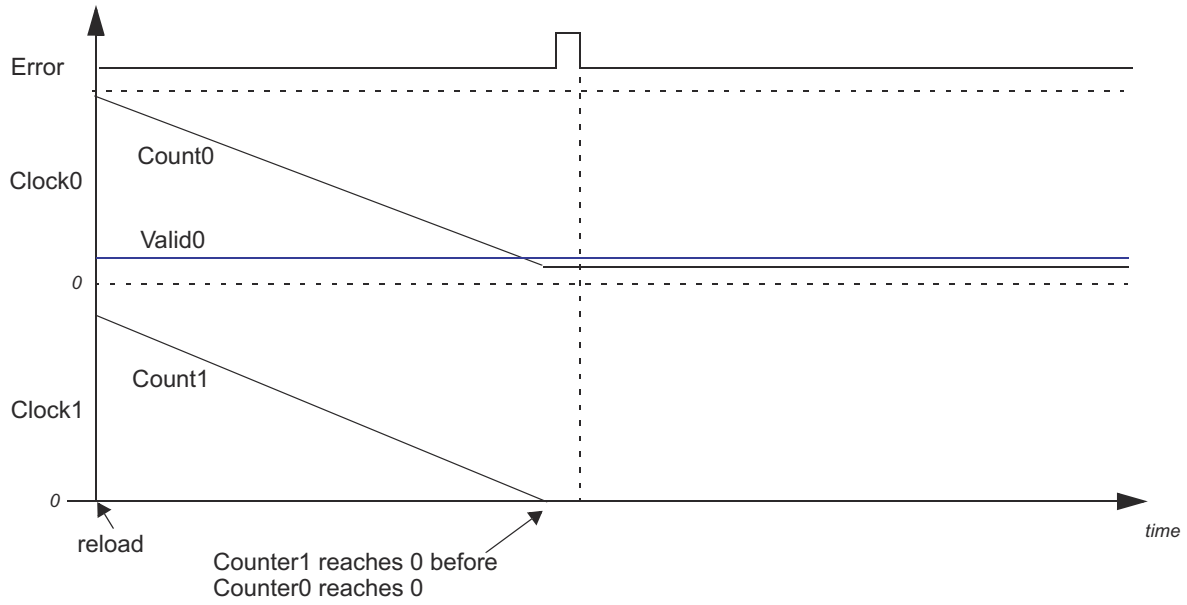


Figure 7-5. Clock1 Faster Than Clock0 - Results in an Error and Stops Counting

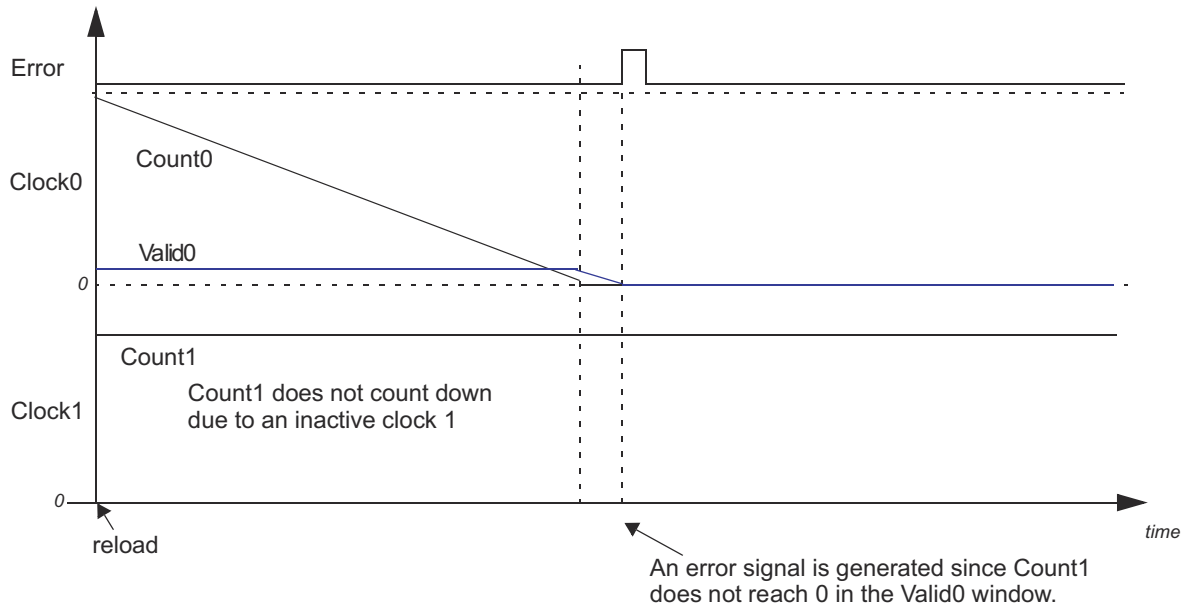


Figure 7-6. Clock1 Not Present - Results in an Error and Stops Counting

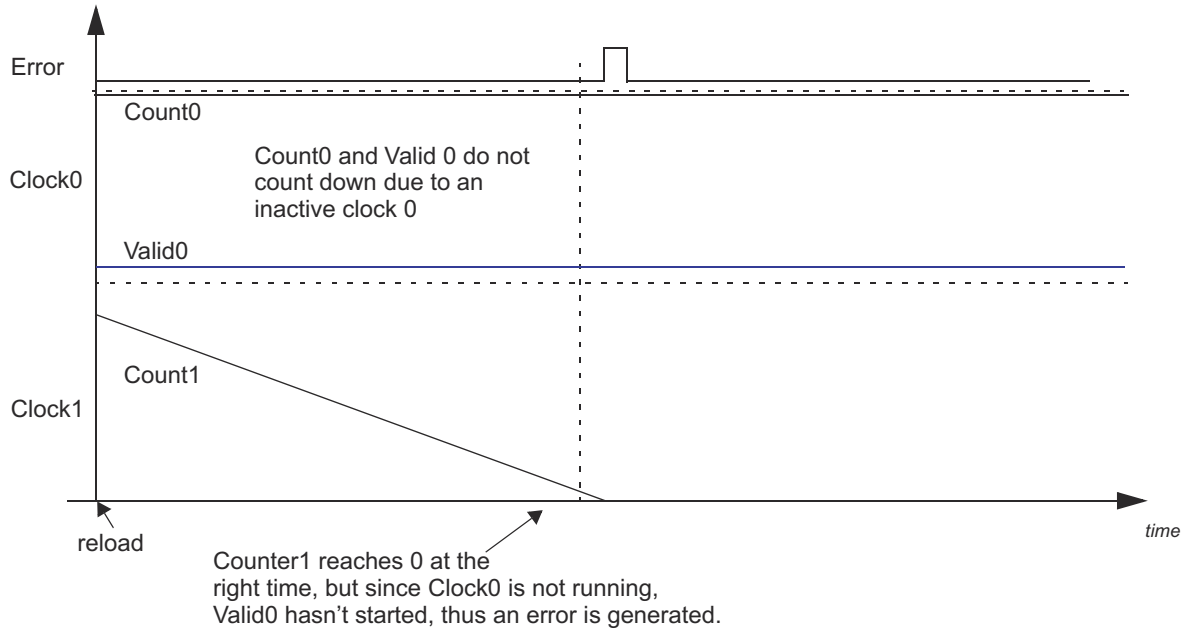


Figure 7-7. Clock0 Not Present - Results in an Error and Stops Counting

7.3 Interrupts

DCC generates an interrupt on either of two events:

- DCC finishes counting and all the counters expire within a defined window indicating DONE operation, provided DCCGCTRL.DONENA = 1.
- DCC finishes counting with error where counters do not expire in a defined window. This indicates an ERROR event, and sets an interrupt provided DCCGCTRL.ERRENA = 1.

Interrupts generated by DONE or ERROR events are ORed and flagged as a SYS_ERR interrupt. Refer to the *PIE Channel Mapping* table in the *System Control and Interrupts* chapter to determine the interrupt channel mapping. The application interrupt service routine needs to check the status flag inside the DCCSTATUS register to determine whether the interrupt is due to ERROR or DONE.

DCC Error interrupts can also be configured as a Non-Maskable Interrupt (NMI) by enabling the CLKFAILCFG.DCCx_ERROR_EN flag.

7.4 Software

7.4.1 DCC Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/dcc

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

7.4.1.1 DCC Single shot Clock verification

FILE: dcc_ex1_single_shot_verification.c

This program uses the XTAL clock as a reference clock to verify the frequency of the PLLRAW clock.

The Dual-Clock Comparator Module 0 is used for the clock verification. The clocksource0 is the reference clock (Fclk0 = 20Mhz) and the clocksource1 is the clock that needs to be verified (Fclk1 = 160Mhz). Seed is the value that gets loaded into the Counter.

Please refer to the TRM for details on counter seed values to be set.

External Connections

- None

Watch Variables

- *status/result* - Status of the PLLRAW clock verification

7.4.1.2 DCC Single shot Clock measurement

FILE: dcc_ex2_single_shot_measurement.c

This program demonstrates Single Shot measurement of the SYSOSCDIV4 clock post trim using XTAL as the reference clock.

The Dual-Clock Comparator Module 0 is used for the clock measurement. The clocksource0 is the reference clock (Fclk0 = 20Mhz) and the clocksource1 is the clock (SYSOSCDIV4) that needs to be measured (Fclk1 = 8 hz). Since the frequency of the clock1 needs to be measured an initial seed is set to the max value of the counter.

Please refer to the TRM for details on counter seed values to be set.

External Connections

- None

- *Watch Variables*
- *result* - Status if the SYSOSCDIV4 clock measurement completed successfully.
- *meas_freq1* - measured clock frequency, in this case for SYSOSCDIV4.

7.4.1.3 DCC Continuous clock monitoring

FILE: dcc_ex3_continuous_monitoring_of_clock.c

This program demonstrates continuous monitoring of PLL Clock in the system using SYSOSCDIV4 as the reference clock. This would trigger an interrupt on any error, causing the decrement/ reload of counters to stop.

The Dual-Clock Comparator Module 0 is used for the clock monitoring. The clocksource0 (SYSOSCDIV4) is the reference clock (Fclk0 = 8 Mhz) and the clocksource1 is the clock that needs to be monitored (Fclk1 = 160Mhz). The clock0 and clock1 seed are set to achieve a window of 400us. Seed is the value that gets loaded into the Counter. For the sake of demo a slight variance is given to clock1 seed value to generate an error on continuous monitoring.

Please refer to the TRM for details on counter seed values to be set. Note : When running in flash configuration it is good to do a reset & restart after loading the example to remove any stale flags/states.

External Connections

- None

Watch Variables

- *status/result* - Status of the PLLRAW clock monitoring
- *cnt0* - Counter0 Value measure when error is generated
- *cnt1* - Counter1 Value measure when error is generated
- *valid* - Valid0 Value measure when error is generated

7.4.1.4 DCC Continuous clock monitoring

FILE: dcc_ex3_continuous_monitoring_of_clock_syscfg.c

This program demonstrates continuous monitoring of PLL Clock in the system using SYSOSCDIV4 as the reference clock. This would trigger an interrupt on any error, causing the decrement/ reload of counters to stop. The Dual-Clock Comparator Module 0 is used for the clock monitoring. The clocksource0 is the reference SYSOSCDIV4 clock (Fclk0 = 8Mhz) and the clocksource1 is the clock that needs to be monitored (Fclk1 = 160Mhz). The clock0 and clock1 seed are set automatically by the error tolerances defined in the sysconfig file included this project. For the sake of demo an un-realistic tolerance is assumed to generate an error on continuous monitoring.

Please refer to the TRM for details on counter seed values to be set. Note : When running in flash configuration it is good to do a reset & restart after loading the example to remove any stale flags/states.

External Connections

- None

Watch Variables

- *status/result* - Status of the PLLRAW clock monitoring
- *cnt0* - Counter0 Value measure when error is generated
- *cnt1* - Counter1 Value measure when error is generated
- *valid* - Valid0 Value measure when error is generated

7.4.1.5 DCC Detection of clock failure

FILE: dcc_ex4_clock_fail_detect.c

This program demonstrates clock failure detection on continuous monitoring of the PLL Clock in the system using XTAL as the osc clock source. Once the oscillator clock fails, it would trigger a DCC error interrupt, causing the decrement/ reload of counters to stop. In this examples, the clock failure is simulated by turning off the XTAL oscillator. Once the ISR is serviced, the osc source is changed to WROSCDIV8 and the PLL is turned off.

The Dual-Clock Comparator Module 0 is used for the clock monitoring. The clocksource0 is the reference clock (Fclk0 = 20Mhz) and the clocksource1 is the clock that needs to be monitored (Fclk1 = 160Mhz). Seed is the value that gets loaded into the Counter.

In the current example, the XTAL is expected to be a Resonator running in Crystal mode which is later switched off to simulate the clock failure. If an SE Crystal is used, you will need to physically disconnect the clock on the board. Please refer to the TRM for details on counter seed values to be set. Note : When running in flash configuration it is good to do a reset & restart after loading the example to remove any stale flags/states.

External Connections

- None

Watch Variables

- *status/result* - Status of the clock failure detection

7.5 DCC Registers

This Section describes the DCC Registers.

7.5.1 DCC Base Address Table

Table 7-1. DCC Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
Dcc0Regs	DCC_REGS	DCC0_BASE	0x0005_E700	-	YES

7.5.2 DCC_REGS Registers

Table 7-2 lists the memory-mapped registers for the DCC_REGS registers. All register offset addresses not listed in Table 7-2 should be considered as reserved locations and the register contents should not be modified.

Table 7-2. DCC_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	DCCGCTRL	Global Control Register	
8h	DCCCNTSEED0	Counter 0 Seed Value	
Ch	DCCVALIDSEED0	Valid 0 Seed Value	
10h	DCCCNTSEED1	Counter 1 Seed Value	
14h	DCCSTATUS	DCC Status	
18h	DCCCNT0	Counter 0 Value	
1Ch	DCCVALID0	Valid Value 0	
20h	DCCCNT1	Counter 1 Value	
24h	DCCCLKSRC1	Clock Source 1	
28h	DCCCLKSRC0	Clock Source 0	

Complex bit access types are encoded to fit into small table cells. Table 7-3 shows the codes that are used for access types in this section.

Table 7-3. DCC_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
R-1	R -1	Read Returns 1s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

7.5.2.1 DCCGCTRL Register (Offset = 0h) [Reset = 00005555h]

DCCGCTRL is shown in Figure 7-8 and described in Table 7-4.

Return to the Summary Table.

Starts / stops the counters. Clears the error signal.

Figure 7-8. DCCGCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DONEENA				SINGLESHOT				ERRENA				DCCENA			
R/W-5h				R/W-5h				R/W-5h				R/W-5h			

Table 7-4. DCCGCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	DONEENA	R/W	5h	DONE Enable Enables/disables the done interrupt signal, but has no effect on the done status flag in DCCSTAT register. 0101 The done signal is disabled Others The done signal is enabled Reset type: SYSRSn
11-8	SINGLESHOT	R/W	5h	Single-Shot Enable Enables/disables repetitive operation of the DCC. 1010: Stop counting when COUNTER0 and VALID0 both reach zero 1011: Reserved Others: Continuously repeat (until error) Reset type: SYSRSn
7-4	ERRENA	R/W	5h	Error Enable Enables/disables the error signal. 0101 The error signal is disabled Others The error signal is enabled Reset type: SYSRSn
3-0	DCCENA	R/W	5h	DCC Enable Starts and stops the operation of the DCC. 0101 Counters are stopped Others Counters are running Reset type: SYSRSn

7.5.2.2 DCCCNTSEED0 Register (Offset = 8h) [Reset = 00000000h]

DCCCNTSEED0 is shown in Figure 7-9 and described in Table 7-5.

Return to the Summary Table.

Seed value for the counter attached to Clock Source 0.

Figure 7-9. DCCCNTSEED0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COUNTSEED0																			
R-0h												R/W-0h																			

Table 7-5. DCCCNTSEED0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	COUNTSEED0	R/W	0h	Seed Value for Counter 0 Contains the seed value that gets loaded into Counter 0 (Clock Source 0). NOTE: Operating the DCC with '0' in the COUNTSEED0 register will result in undefined operation. Reset type: SYSRSn

7.5.2.3 DCCVALIDSEED0 Register (Offset = Ch) [Reset = 00000000h]

DCCVALIDSEED0 is shown in Figure 7-10 and described in Table 7-6.

Return to the Summary Table.

Seed value for the timeout counter attached to Clock Source 0.

Figure 7-10. DCCVALIDSEED0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VALIDSEED															
R-0h																R/W-0h															

Table 7-6. DCCVALIDSEED0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	VALIDSEED	R/W	0h	Seed Value for Valid Duration Counter 0 Contains the seed value that gets loaded into the valid duration counter for Clock Source 0. NOTE: Operating the DCC with '0' in the VALIDSEED0 register will result in undefined operation. VALID0 defines a window in which COUNT1 expires. This window is meant to be at least four cycles wide. Do not program a value less than '4' into the VALID0 register. Reset type: SYSRSn

7.5.2.4 DCCCNTSEED1 Register (Offset = 10h) [Reset = 00000000h]

DCCCNTSEED1 is shown in Figure 7-11 and described in Table 7-7.

Return to the Summary Table.

Seed value for the counter attached to Clock Source 1.

Figure 7-11. DCCCNTSEED1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COUNTSEED1																			
R-0h												R/W-0h																			

Table 7-7. DCCCNTSEED1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	COUNTSEED1	R/W	0h	Seed Value for Counter 1 Contains the seed value that gets loaded into Counter 1 (Clock Source 1). NOTE: Operating the DCC with '0' in the COUNTSEED1 register will result in undefined operation. Reset type: SYSRSn

7.5.2.5 DCCSTATUS Register (Offset = 14h) [Reset = 00000000h]

DCCSTATUS is shown in Figure 7-12 and described in Table 7-8.

Return to the Summary Table.

Specifies the status of the DCC Module.

Figure 7-12. DCCSTATUS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						DONE	ERR
R-0h						R/W-0h	R/W-0h

Table 7-8. DCCSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	DONE	R/W	0h	Single-Shot Done Flag Indicates when single-shot mode is complete without error. Writing a '1' to this bit clears the flag. 0 Single-shot mode has not completed. 1 Single-shot mode has completed. Reset type: SYSRSn
0	ERR	R/W	0h	Error Flag Indicates whether or not an error has occurred. Writing a '1' to this bit clears the flag. 0 No errors have occurred. 1 An error has occurred. Reset type: SYSRSn

7.5.2.6 DCCCNT0 Register (Offset = 18h) [Reset = 00000000h]

DCCCNT0 is shown in Figure 7-13 and described in Table 7-9.

Return to the Summary Table.

Value of the counter attached to Clock Source 0.

Figure 7-13. DCCCNT0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COUNT0																			
R-0h												R-0h																			

Table 7-9. DCCCNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	COUNT0	R	0h	Current Value of Counter 0 Reset type: SYSRSn

7.5.2.7 DCCVALID0 Register (Offset = 1Ch) [Reset = 00000000h]

DCCVALID0 is shown in Figure 7-14 and described in Table 7-10.

Return to the Summary Table.

Value of the valid counter attached to Clock Source 0.

Figure 7-14. DCCVALID0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VALID0															
R-0h																R-0h															

Table 7-10. DCCVALID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	VALID0	R	0h	Current Value of Valid 0 Reset type: SYSRSn

7.5.2.8 DCCCNT1 Register (Offset = 20h) [Reset = 00000000h]

DCCCNT1 is shown in Figure 7-15 and described in Table 7-11.

Return to the Summary Table.

Value of the counter attached to Clock Source 1.

Figure 7-15. DCCCNT1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COUNT1																			
R-0h												R-0h																			

Table 7-11. DCCCNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-0	COUNT1	R	0h	Current Value of Counter 1 Reset type: SYSRSn

7.5.2.9 DCCCLKSRC1 Register (Offset = 24h) [Reset = 00000000h]

DCCCLKSRC1 is shown in Figure 7-16 and described in Table 7-12.

Return to the Summary Table.

Selects the clock source for Counter 1.

Figure 7-16. DCCCLKSRC1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY				RESERVED						CLKSRC1					
R-0/W-0h				R-0h						R/W-0h					

Table 7-12. DCCCLKSRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	KEY	R-0/W	0h	Enables or Disables Clock Source Write for COUNT1 1010 The CLKSRC field selects the clock source for COUNT1. Others: Previous values retained new writes on register fields has no impact. Reset type: SYSRSn
11-6	RESERVED	R	0h	Reserved
5-0	CLKSRC1	R/W	0h	Clock Source Select for Counter 1 Specifies the clock source for COUNT1, when the KEY field enables this feature. Note: Any values not explicitly defined below are reserved. Reset type: SYSRSn 0h (R/W) = Direct output of SYSPLL CLKOUT 1h (R/W) = Reserved 2h (R/W) = WROSC output clock after the fixed BY8 divider 3h (R/W) = SYSOSC output clock after the fixed BY4 divider 4h (R/W) = Reserved 5h (R/W) = Reserved 6h (R/W) = CPU1 system clock. 7h (R/W) = Reserved 8h (R/W) = Reserved 9h (R/W) = Input 15 of INPUTXBAR1 Ah (R/W) = Reserved Bh (R/W) = Clock input to MCPWM module Ch (R/W) = Bit clock for SPI and SCI modules Dh (R/W) = ADC conversion clock Eh (R/W) = Watchdog clock after dividers Fh (R/W) = Reserved 10h (R/W) = Reserved 11h (R/W) = Reserved 12h (R/W) = Reserved 13h (R/W) = Reserved 14h (R/W) = Reserved 15h (R/W) = Reserved 16h (R/W) = Reserved 17h (R/W) = FCLK (divided clock) output from Flash wrapper 18h (R/W) = Input 11 of INPUTXBAR1 19h (R/W) = Input 12 of INPUTXBAR1 1Ah (R/W) = Reserved 1Bh (R/W) = Reserved 1Ch (R/W) = Reserved 1Dh (R/W) = Reserved 1Eh (R/W) = Reserved

7.5.2.10 DCCCLKSRC0 Register (Offset = 28h) [Reset = 00000000h]

DCCCLKSRC0 is shown in Figure 7-17 and described in Table 7-13.

Return to the Summary Table.

Selects the clock source for Counter 0.

Figure 7-17. DCCCLKSRC0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY				RESERVED							CLKSRC0				
R-0/W-0h				R-0h							R/W-0h				

Table 7-13. DCCCLKSRC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-12	KEY	R-0/W	0h	Enables or Disables Clock Source Write for COUNT0 1010: The CLKSRC0 field written with key gets updated to with new selection to clock COUNT0. Others: Previous values retained new writes on register fields has no impact. Reset type: SYSRSn
11-5	RESERVED	R	0h	Reserved
4-0	CLKSRC0	R/W	0h	Clock Source Select for Counter 0 Specifies the clock source for COUNT0, when the KEY field enables this feature. Note: All values not defined below are reserved. Reset type: SYSRSn 0h (R/W) = Crystal oscillator output 1h (R/W) = WROSC output clock after the fixed BY8 divider 2h (R/W) = SYSOSC output clock after the fixed BY4 divider 4h (R/W) = TCK pin input 5h (R/W) = CPU1 system clock 8h (R/W) = Reserved Ch (R/W) = Input 16 of INPUTXBAR1 Eh (R/W) = Reserved Fh (R/W) = Reserved

Chapter 8

General-Purpose Input/Output (GPIO)

The GPIO module controls the device's digital multiplexing, which uses shared pins to maximize application flexibility. The pins are named by the general-purpose I/O name (for example, GPIO0, GPIO25, GPIO58). These pins can be individually selected to operate as digital I/O (also called GPIO mode), or connected to one of several peripheral I/O signals. The input signals can be qualified to remove unwanted noise.

8.1 Introduction

Up to twelve independent peripheral signals are multiplexed on a single GPIO-enabled pin in addition to the CPU-controlled I/O capability. Each pin output can be controlled by either a peripheral or one of the CPU controllers.

- CPU1

There are up to 8 possible I/O ports:

- Port A consists of GPIO0-GPIO31
- Port B consists of GPIO32-GPIO63
- Port C consists of GPIO64-GPIO95
- Port D consists of GPIO96-GPIO127
- Port E consists of GPIO128-GPIO159
- Port F consists of GPIO160-GPIO191
- Port G consists of GPIO192-GPIO223
- Port H consists of GPIO224-GPIO255

Note

Some GPIO and I/O ports can be unavailable on particular devices. See the *GPIO Registers* section for available GPIO and I/O ports.

The analog signals on this device are multiplexed with digital inputs and outputs. Some of these analog IO (AIO) pins do not have digital output capability. Others of these pins are analog pins capable of full digital input and output capability (AGPIO). Analog pins with AIO (digital input only) capability contain "AIO" signals in the Pin Attributes table of the device data sheet. Analog pins with full input and output capability (AGPIO pins) contain "GPIO" signals in the Pin Attributes table of the device data sheet. AGPIO pins also have pin names with both analog signals and GPIO in the name.

Figure 8-1 shows the GPIO logic for a single pin.

There are two key features to note in Figure 8-1. The first is that the input and output paths are entirely separate, connecting only at the pin. The second is that peripheral muxing takes place far from the pin. As a result, for the CPU to read the physical state of the pin independent of peripheral muxing is possible. Likewise, external interrupts can be generated from peripheral activity. All pin options such as input qualification and open-drain output are valid for all peripherals.

Note

JTAG uses a different signal path that does not support inversion or qualification.

GPIO18/X2 and GPIO19/X1 have different timings due to the load placed on them by the oscillator circuit. For information on using GPIO18/X2 and GPIO19/X1 as GPIOs, see the data sheet and the *Clocking* section of this document.

If digital signals with sharp edges (high dv/dt) are connected to the AIOs or AGPIOs, cross-talk can occur with adjacent analog signals. Therefore, limit the edge rate of signals connected to AIOs or AGPIOs if adjacent channels are being used for analog functions.

Note

In open-drain mode, the GPIO does not drive the pin high, the GPIO can only pull the pin low. Instead, use an external pull-up to the bus voltage to drive the high level. When open-drain mode is enabled, the value in the GPyDAT register still controls the pin state. Writing a value of 1 turns off the driver to allow the external pull-up to control the pin; writing a value of 0 pulls the pin to ground. The open-drain configuration is automatically used by peripherals such as I2C and PMBus (no need to enable open-drain mode locally). This mode can also be set manually by writing to the GPyODR register and can be used when there are multiple nodes on the same net to avoid the pin contention that a push-pull driver can cause.

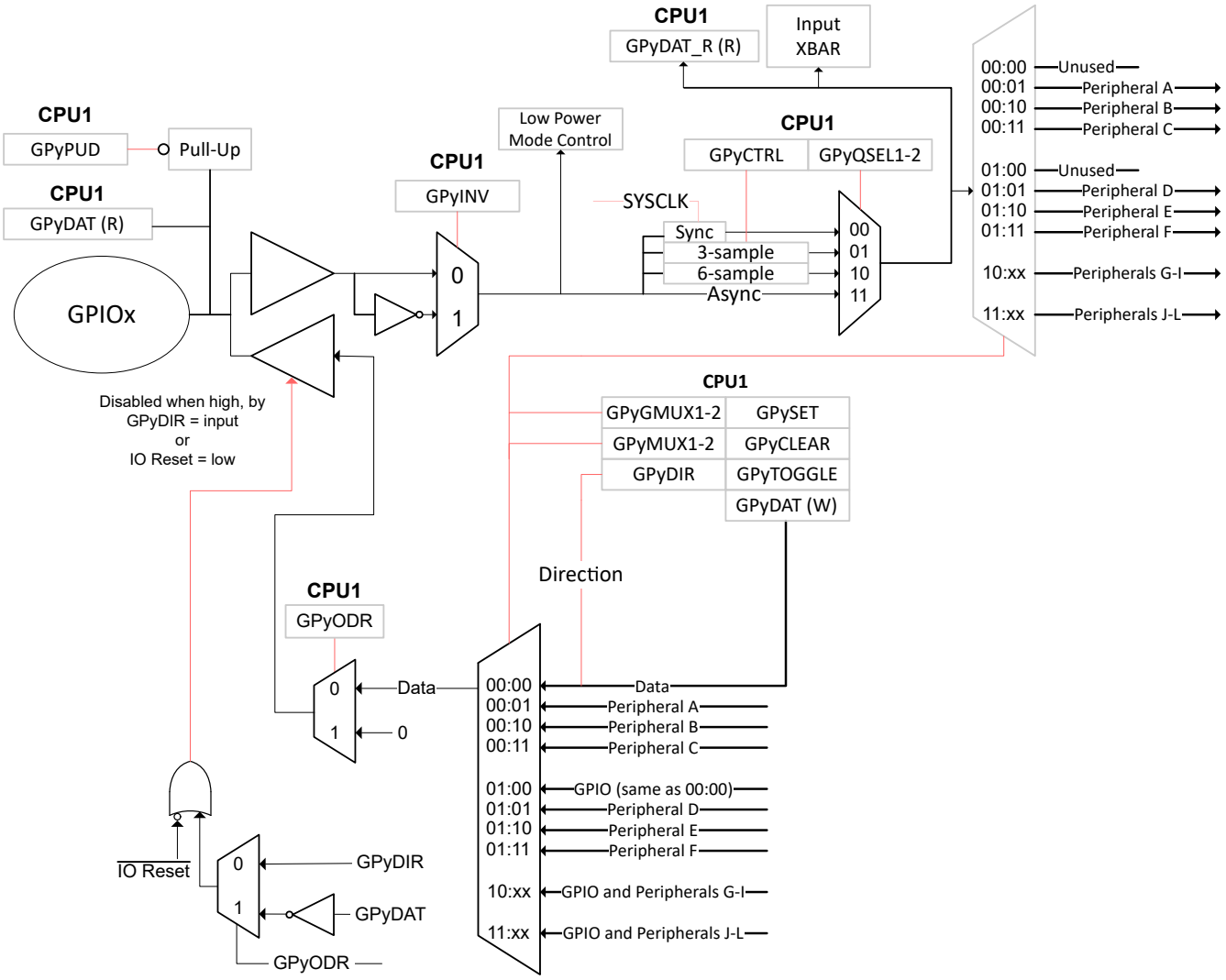


Figure 8-1. GPIO Logic for a Single Pin

8.1.1 GPIO Related Collateral

Foundational Materials

- C28x Academy - GPIO

Getting Started Materials

- *How to Maximize GPIO Usage in C2000 Devices Application Report*
- [FAQ] C2000 GPIO FAQ

8.2 Configuration Overview

I/O pin configuration consists of several steps:

1. **Plan the device pin-out:** Make a list of all required peripherals for the application. Using the peripheral mux information in the device data sheet, choose which GPIOs to use for the peripheral signals. Decide which of the remaining GPIOs to use as inputs and outputs.

   Once the peripheral muxing has been chosen, implement the mux by writing the appropriate values to the GPyMUX1/2 and GPyGMUX1/2 registers. When changing the GPyGMUX value for a pin, always set the corresponding GPyMUX bits to zero first to avoid glitching in the muxes. By default, all pins are general-purpose I/Os, not peripheral signals, with the exception of GPIO35 and GPIO37.
2. **(Optional) Enable internal pullup resistors:** To enable or disable the pullup resistors, write to the appropriate bits in the GPIO pullup disable registers (GPyPUD). All pullups are disabled by default. Pullups can be used to keep input pins in a known state when there is no external signal driving them.
3. **Select input qualification:** If the pin is used as an input, specify the required input qualification, if any. The input qualification sampling period is selected in the GPyCTRL registers, while the type of qualification is selected in the GPyQSEL1 and GPyQSEL2 registers. By default, all qualification is synchronous with a sampling period equal to PLLSYSCLK, with the exception of GPIO35 and GPIO37. For an explanation of input qualification, see Section 8.6.
4. **Select the direction of any general-purpose I/O pins:** For each pin configured as a GPIO, specify the direction of the pin as either input or output using the GPyDIR registers. By default, all GPIO pins are inputs. Before changing a pin to an output, load the output latch with the value to be driven by writing that value to the GPySET, GPyCLEAR, or GPyDAT registers. Once the latch is loaded, write to GPyDIR to change the pin direction. By default, all output latches are zero.

   The GPyDAT_R register can be used to read what value was written to the GPyDAT register.
5. **Select low-power mode wake-up sources:** GPIOs 0-63 can be used to wake the system up from low power modes. To select one or more GPIOs for wake-up, write to the appropriate bits in the GPIOLPMSEL0 and GPIOLPMSEL1 registers. These registers are part of the CPU system register space. For more information on low-power modes and GPIO wake-up, see the Low-Power Modes section in the *System Control and Interrupts* chapter.
6. **Select external interrupt sources:** Configuring external interrupts is a two-step process. First, the interrupts themselves must be enabled and the polarity must be configured using the XINTnCR registers. Second, the XINT1-5 GPIO pins must be set by selecting the sources for Input X-BAR signals 4, 5, 6, 13, and 14, respectively. For more information on the Input X-BAR architecture, see the *Crossbar (X-BAR)* chapter.

8.3 Digital Inputs on ADC Pins (AIOs)

Some GPIOs are multiplexed with analog pins and only have digital input functionality. These are also referred to as AIOs. Pins with only an AIO option on this port can only function in input mode. See the device data sheet for list of AIO signals. By default, these pins function as analog pins and the GPIOs are in a high-impedance state. The GPyAMSEL register is used to configure these pins for digital or analog operation.

Note

If digital signals with sharp edges (high dv/dt) are connected to the AIOs, cross-talk can occur with adjacent analog signals. Therefore, limit the edge rate of signals connected to AIOs if adjacent channels are being used for analog functions.

8.4 Digital Inputs and Outputs on ADC Pins (AGPIOs)

Some GPIOs are multiplexed with analog pins and have digital input and output functionality. These are also referred to as AGPIOs. Unlike AIOs, AGPIOs have full input and output capability. By default, the AGPIOs are not connected and must be configured. Table 8-1 shows how to configure the AGPIOs. To enable the analog functionality, set the register AGPIOCTRLx from analog subsystem. To enable the digital functionality, set the register GPxAMSEL from the *General-Purpose Input/Output (GPIO)* chapter.

Table 8-1. AGPIO Configuration

AGPIOCTRLx.GPIOy (Default = 0)	GPxAMSEL.GPIOy (Default = 1)	Pin Connected To:	
		ADC	GPIOy
0	0	-	Yes
0	**1**	**-** (1)	**-** (1)
1	0	-	Yes
1	1	Yes	-

(1) By default there are no signals connected to AGPIO pins. One of the other rows in the table must be chosen for pin functionality.

Note

If digital signals with sharp edges (high dv/dt) are connected to the AGPIOs, cross-talk can occur with adjacent analog signals. The user must therefore limit the edge rate of signals connected to AGPIOs, if adjacent channels are being used for analog functions.

8.5 Digital General-Purpose I/O Control

The values on the pins that are configured as GPIO can be changed by using the following registers.

- **GPyDAT Registers**

  Each I/O port has one data register. Each bit in the data register corresponds to one GPIO pin. No matter how the pin is configured (GPIO or peripheral function), the corresponding bit in the data register reflects the current state of the pin after qualification. Writing to the GPyDAT register clears or sets the corresponding output latch and if the pin is enabled as a general-purpose output (GPIO output), the pin is also driven either low or high. If the pin is not configured as a GPIO output, then the value is latched but the pin is not driven. Only if the pin is later configured as a GPIO output is the latched value driven onto the pin.

  When using the GPyDAT register to change the level of an output pin, be cautious to not accidentally change the level of another pin. For example, to change the output latch level of GPIOA0 by writing to the GPADAT register bit 0 using a read-modify-write instruction, a problem can occur if another I/O port A signal changes level between the read and the write stage of the instruction. Following is an analysis of why this happens:

  The GPyDAT registers reflect the state of the pin, not the latch. This means the register reflects the actual pin value. However, there is a lag between when the register is written to when the new pin value is reflected back in the register. This can pose a problem when this register is used in subsequent program statements to alter the state of GPIO pins. An example is shown below where two program statements attempt to drive two different GPIO pins that are currently low to a high state.

  If Read-Modify-Write operations are used on the GPyDAT registers, because of the delay between the output and the input of the first instruction (I1), the second instruction (I2) reads the old value and writes the value back.

```
GpioDataRegs.GPADAT.bit.GPIO1 = 1;  //I1 performs read-modify-write of GPADAT
GpioDataRegs.GPADAT.bit.GPIO2 = 1;  //I2 also a read-modify-write of GPADAT
                                    //GPADAT gets the old value of GPIO1 due to the delay
```

  The second instruction waits for the first to finish the write due to the write-followed-by-read protection on this peripheral frame. There is some lag, however, between the write of (I1) and the GPyDAT bit reflecting the new value (1) on the pin. During this lag, the second instruction reads the old value of GPIO1 (0) and writes the value back along with the new value of GPIO2 (1). Therefore, GPIO1 pin stays low.

  One answer is to put some NOPs between instructions. A better answer is to use the GPySET/GPyCLEAR/GPyTOGGLE registers instead of the GPyDAT registers. These registers always read back a 0 and writes of 0 have no effect. Only bits that need to be changed can be specified without disturbing any other bits that are currently in the process of changing.

- **GPyDAT_R Registers**

  The GPyDAT_R registers are read only registers that return the value written to the GPyDAT registers instead of pin status. Writes to these registers have no effect.

- **GPySET Registers**

  The set registers are used to drive specified GPIO pins high without disturbing other pins. Each I/O port has one set register and each bit corresponds to one GPIO pin. The set registers always read back 0. If the corresponding pin is configured as an output, then writing a 1 to that bit in the set register sets the output latch high and the corresponding pin is driven high. If the pin is not configured as a GPIO output, then the value is latched but the pin is not driven. Only if the pin is later configured as a GPIO output is the latched value driven onto the pin. Writing a 0 to any bit in the set registers has no effect.

- **GPyCLEAR Registers**

  The clear registers are used to drive specified GPIO pins low without disturbing other pins. Each I/O port has one clear register. The clear registers always read back 0. If the corresponding pin is configured as a general-purpose output, then writing a 1 to the corresponding bit in the clear register clears the output latch and the pin is driven low. If the pin is not configured as a GPIO output, then the value is latched but the pin is not driven. Only if the pin is later configured as a GPIO output is the latched value driven onto the pin. Writing a 0 to any bit in the clear registers has no effect.

- **GPyTOGGLE Registers**

  The toggle registers are used to drive specified GPIO pins to the opposite level without disturbing other pins. Each I/O port has one toggle register. The toggle registers always read back 0. If the corresponding pin is configured as an output, then writing a 1 to that bit in the toggle register flips the output latch and pulls the corresponding pin in the opposite direction. That is, if the output pin is driven low, then writing a 1 to the corresponding bit in the toggle register pulls the pin high. Likewise, if the output pin is high, then writing a 1 to the corresponding bit in the toggle register pulls the pin low. If the pin is not configured as a GPIO output, then the value is latched but the pin is not driven. Only if the pin is later configured as a GPIO output is the latched value driven onto the pin. Writing a 0 to any bit in the toggle registers has no effect.

8.6 Input Qualification

The input qualification scheme has been designed to be very flexible. Select the type of input qualification for each GPIO pin by configuring the GPyQSEL1 and GPyQSEL2 registers. In the case of a GPIO input pin, the qualification can be specified as only synchronized to SYSCLKOUT or qualification by a sampling window. For pins that are configured as peripheral inputs, the input can also be asynchronous in addition to synchronized to SYSCLKOUT or qualified by a sampling window. The remainder of this section describes the options available.

8.6.1 No Synchronization (Asynchronous Input)

This mode is used for peripherals where input synchronization is not required or the peripheral performs the synchronization. Examples include communication ports McBSP, SCI, SPI, and I^2C. In addition, the MCPWM trip zone ($\overline{TZn}$) signals can function independent of the presence of SYSCLKOUT.

Note

Using input synchronization when the peripheral performs the synchronization can cause unexpected results. The user must make sure that the GPIO pin is configured for asynchronous in this case.

8.6.2 Synchronization to SYSCLKOUT Only

This is the default qualification mode of all the pins at reset. In this mode, the input signal is only synchronized to the system clock (SYSCLKOUT). Because the incoming signal is asynchronous, a SYSCLKOUT period of delay is needed for the input to the device to be changed. No further qualification is performed on the signal.

8.6.3 Qualification Using a Sampling Window

In this mode, the signal is first synchronized to the system clock (SYSCLKOUT) and then qualified by a specified number of cycles before the input is allowed to change. Figure 8-2 and Figure 8-3 show how the input qualification is performed to eliminate unwanted noise. Two parameters are specified by the user for this type of qualification: 1) the sampling period, or how often the signal is sampled, and 2) the number of samples to be taken.

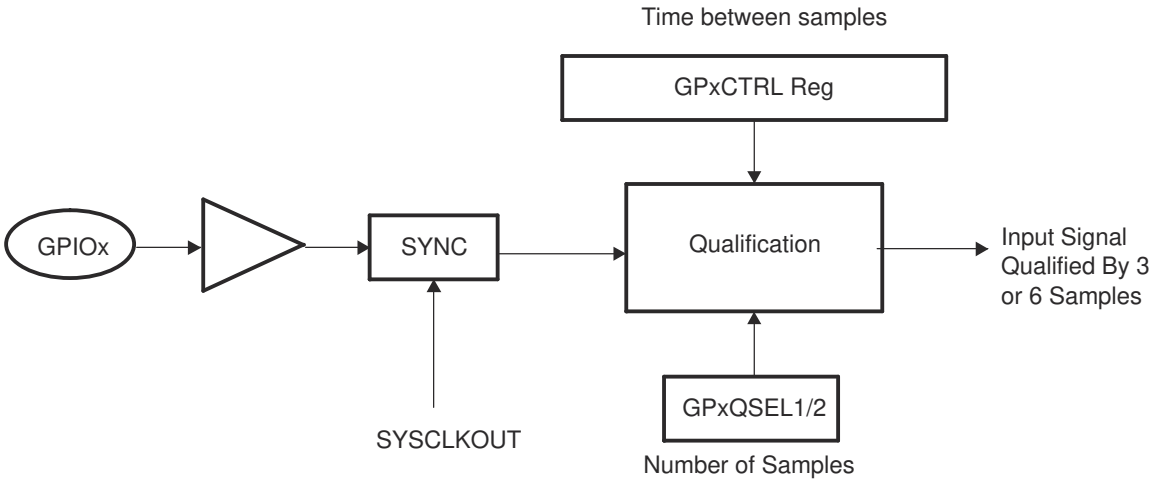


Figure 8-2. Input Qualification Using a Sampling Window

Time between samples (sampling period):

To qualify the signal, the input signal is sampled at a regular period. The sampling period is specified by the user and determines the time duration between samples, or how often the signal is sampled, relative to the CPU clock (SYSCLKOUT).

The sampling period is specified by the qualification period (QUALPRDn) bits in the GPxCTRL register . The sampling period is configurable in groups of 8 input signals. For example, GPIO0 to GPIO7 use GPACTRL[QUALPRD0] setting and GPIO8 to GPIO15 use GPACTRL[QUALPRD1]. Table 8-2 and Table 8-3 show the relationship between the sampling period or sampling frequency and the GPxCTRL[QUALPRDn] setting.

Table 8-2. Sampling Period

	Sampling Period
If GPxCTRL[QUALPRDn] = 0	$1 \times T_{SYSCLKOUT}$
If GPxCTRL[QUALPRDn] ≠ 0	$2 \times$ GPxCTRL[QUALPRDn] $\times T_{SYSCLKOUT}$
	Where $T_{SYSCLKOUT}$ is the period in time of SYSCLKOUT

Table 8-3. Sampling Frequency

	Sampling Frequency
If GPxCTRL[QUALPRDn] = 0	$f_{SYSCLKOUT}$
If GPxCTRL[QUALPRDn] ≠ 0	$f_{SYSCLKOUT} \times 1 \div (2 \times$ GPxCTRL[QUALPRDn])
	Where $f_{SYSCLKOUT}$ is the frequency of SYSCLKOUT

From these equations, the minimum and maximum time between samples can be calculated for a given SYSCLKOUT frequency:

Example: Maximum Sampling Frequency:

If GPxCTRL[QUALPRDn] = 0

then the sampling frequency is $f_{SYSCLKOUT}$

If, for example, $f_{SYSCLKOUT}$ = 60MHz

then the signal is sampled at 60MHz or one sample every 16.67ns.

Example: Minimum Sampling Frequency:

If GPxCTRL[QUALPRDn] = 0xFF (255)

then the sampling frequency is $f_{SYSCLKOUT} \times 1 \div (2 \times$ GPxCTRL[QUALPRDn])

If, for example, $f_{SYSCLKOUT}$ = 60MHz

then the signal is sampled at 60MHz × 1 ÷ (2 × 255) (117.647kHz) or one sample every 8.5µs.

Number of samples:

The number of times the signal is sampled is either three samples or six samples as specified in the qualification selection (GPyQSEL1, GPyQSEL2) registers. When three or six consecutive cycles are the same, then the input change is passed through to the device.

Total Sampling-Window Width:

The sampling window is the time during which the input signal is sampled as shown in Figure 8-3. By using the equation for the sampling period, along with the number of samples to be taken, the total width of the window can be determined.

For the input qualifier to detect a change in the input, the level of the signal must be stable for the duration of the sampling-window width or longer.

The number of sampling periods within the window is always one less than the number of samples taken. For a three-sample window, the sampling-window width is two sampling-periods wide where the sampling period is defined in Table 8-2. Likewise, for a six-sample window, the sampling-window width is five sampling-periods wide. Table 8-4 and Table 8-5 show the calculations used to determine the total sampling-window width based on GPxCTRL[QUALPRDn] and the number of samples taken.

Table 8-4. Case 1: Three-Sample Sampling-Window Width

	Total Sampling-Window Width
If GPxCTRL[QUALPRDn] = 0	2 × $T_{SYSCLKOUT}$
If GPxCTRL[QUALPRDn] ≠ 0	2 × 2 × GPxCTRL[QUALPRDn] × $T_{SYSCLKOUT}$
	Where $T_{SYSCLKOUT}$ is the period in time of SYSCLKOUT

Table 8-5. Case 2: Six-Sample Sampling-Window Width

	Total Sampling-Window Width
If GPxCTRL[QUALPRDn] = 0	5 × $T_{SYSCLKOUT}$
If GPxCTRL[QUALPRDn] ≠ 0	5 × 2 × GPxCTRL[QUALPRDn] × $T_{SYSCLKOUT}$
	Where $T_{SYSCLKOUT}$ is the period in time of SYSCLKOUT

Note

The external signal change is asynchronous with respect to both the sampling period and SYSCLKOUT. Due to the asynchronous nature of the external signal, the input must be held stable for a time greater than the sampling-window width to make sure the logic detects a change in the signal. The extra time required can be up to an additional sampling period + $T_{SYSCLKOUT}$.

The required duration for an input signal to be stable for the qualification logic to detect a change is described in the data sheet.

Example Qualification Window:

For the example shown in Figure 8-3, the input qualification has been configured as follows:

- GPxQSEL1/2 = 1,0. This indicates a six-sample qualification.
- GPxCTRL[QUALPRDn] = 1. The sampling period is $t_w(SP) = 2 \times$ GPxCTRL[QUALPRDn] $\times T_{SYSCLKOUT} = 2 \times T_{SYSCLKOUT}$.

This configuration results in the following:

- The width of the sampling window is:

  $t_w(IQSW) = 5 \times t_w(SP) = 5 \times 2 \times$ GPxCTRL[QUALPRDn] $\times T_{SYSCLKOUT} = 5 \times 2 \times T_{SYSCLKOUT}$

- If, for example, $T_{SYSCLKOUT}$ = 16.67ns, then the duration of the sampling window is:

  Sampling period, $t_w(SP) = 2 \times T_{SYSCLKOUT} = 2 \times 16.67ns = 33.3ns$

  Sampling window, $t_w(IQSW) = 5 \times t_w(SP) = 5 \times 33.3ns = 166.7ns$

- To account for the asynchronous nature of the input relative to the sampling period and SYSCLKOUT, up to a single additional sampling period and SYSCLK period is required to detect a change in the input signal. For this example:

  $t_w(IQSW) + t_w(SP) + T_{SYSCLKOUT} = 166.7ns + 33.3ns + 16.67ns = 216.7ns$

- In Figure 8-3, the glitch (A) is shorter then the qualification window and is ignored by the input qualifier.

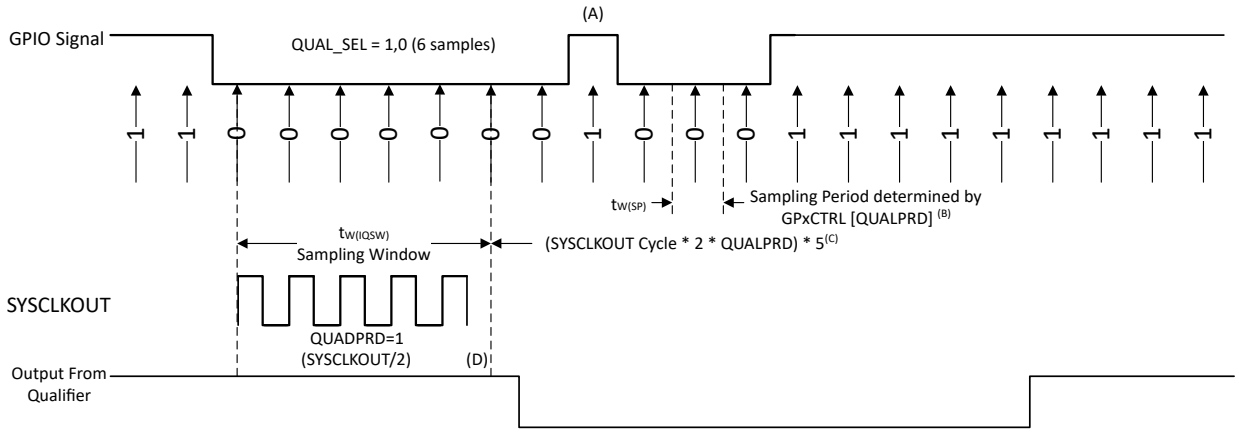


Figure 8-3. Input Qualifier Clock Cycles

- **A.** This glitch is ignored by the input qualifier. The QUALPRD bit field specifies the qualification sampling period and can vary from 0x00 to 0xFF. If QUALPRD = 00, then the sampling period is 1 SYSCLKOUT cycle. For any other value 'n', the qualification sampling period is 2n SYSCLKOUT cycles (that is, at every 2n SYSCLKOUT cycles, the GPIO pin is sampled).
- **B.** The qualification period selected using the GPxCTRL register applies to groups of 8 GPIO pins.
- **C.** The qualification block can take either 3 or 6 samples. The QUAL_SEL Register selects which samples mode is used.
- **D.** In the example shown, for the qualifier to detect the change, the input must be stable for 10 SYSCLKOUT cycles. That makes sure of 5 sampling periods for detection to occur. Since external signals are driven asynchronously, a 13 SYSCLKOUT-wide pulse makes sure of reliable recognition.

8.7 GPIO and Peripheral Muxing

8.7.1 GPIO Muxing

Up to twelve different peripheral functions are multiplexed to each pin along with a general-purpose input/output (GPIO) function. This allows you to choose the peripheral mix and pinout that works best for your particular application. Refer to for muxing combinations and definitions.

Table 8-6. GPIO Muxed Pins

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO0	MCPWM1_1A		OUTPUTXBAR7	SCIA_RX	I2CA_SDA	SPIA_PTE				EQEP1_INDEX		MCPWM1_3A	
GPIO1	MCPWM1_1B		OUTPUTXBAR4	SCIA_TX	I2CA_SCL	SPIA_POCI	EQEP1_STROBE					MCPWM1_3B	
GPIO2	MCPWM1_2A			OUTPUTXBAR1		SPIA_PICO	SCIA_TX		I2CA_SDA				
GPIO3	MCPWM1_2B	OUTPUTXBAR2		OUTPUTXBAR2		SPIA_CLK	SCIA_RX		I2CA_SCL				
GPIO4	MCPWM1_3A	I2CA_SCL		OUTPUTXBAR3			EQEP1_STROBE				SPIA_POCI	MCPWM1_1A	
GPIO5	MCPWM1_3B	I2CA_SDA	OUTPUTXBAR3			SPIA_PTE	SPIA_POCI		SCIA_RX			MCPWM1_1B	
GPIO6		OUTPUTXBAR4	SYNCOUT	EQEP1_A				MCPWM1_3A				MCPWM1_2A	
GPIO7		MCPWM1_2A	OUTPUTXBAR5	EQEP1_B		SPIA_PICO	MCPWM3_1A		SCIA_TX			MCPWM1_2B	
GPIO8			ADCSOCAO	EQEP1_STROBE	SCIA_TX	SPIA_PICO	I2CA_SCL						
GPIO9		SCIB_TX	OUTPUTXBAR6	EQEP1_INDEX	SCIA_RX	SPIA_CLK	MCPWM1_1B				I2CA_SCL		
GPIO10		MCPWM1_2B	ADCSOCBO	EQEP1_A	SCIB_TX	SPIA_POCI	I2CA_SDA						
GPIO11			OUTPUTXBAR7	EQEP1_B	SCIB_RX	SPIA_PTE	MCPWM3_1B		EQEP1_A	SPIA_PICO			
GPIO12	MCPWM3_1A			EQEP1_STROBE	SCIB_TX				SPIA_CLK				
GPIO13	MCPWM3_1B			EQEP1_INDEX	SCIB_RX				SPIA_POCI				
GPIO16	SPIA_PICO		OUTPUTXBAR7		SCIA_TX		EQEP1_STROBE		XCLKOUT	EQEP1_B			
GPIO17	SPIA_POCI		OUTPUTXBAR8		SCIA_RX		EQEP1_INDEX						
GPIO18	SPIA_CLK	SCIB_TX			I2CA_SCL		EQEP1_A		XCLKOUT				X2
GPIO19	SPIA_PTE	SCIB_RX			I2CA_SDA		EQEP1_B						X1
GPIO20	EQEP1_A				SPIA_PICO				I2CA_SCL			UARTA_TX	
GPIO21	EQEP1_B				SPIA_POCI				I2CA_SDA			UARTA_RX	
GPIO22	EQEP1_STROBE		SCIB_TX										
GPIO23	EQEP1_INDEX	SPIA_PTE	SCIB_RX										
GPIO24	OUTPUTXBAR1	EQEP1_A	SPIA_PTE		SPIA_PICO				SCIA_TX	ERRORSTS			

Table 8-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO28	SCIA_RX	OUTPUTXBAR8	MCPWM3_1A	OUTPUTXBAR5	EQEP1_A		EQEP1_STROBE	UARTA_TX	SPIA_CLK	ERRORSTS	I2CA_SDA		
GPIO29	SCIA_TX	MCPWM1_2A	MCPWM3_1B	OUTPUTXBAR6	EQEP1_B		EQEP1_INDEX	UARTA_RX	SPIA_PTE	ERRORSTS	I2CA_SCL		
GPIO30				OUTPUTXBAR7	EQEP1_STROBE				MCPWM1_1A				
GPIO32	I2CA_SDA	EQEP1_INDEX	SPIA_CLK		UARTA_RX					ADCSOCBO			
GPIO33	I2CA_SCL			OUTPUTXBAR4	UARTA_TX				EQEP1_B	ADCSOCAO			
GPIO35	SCIA_RX	SPIA_POCI	I2CA_SDA			UARTA_RX	EQEP1_A					TDI	
GPIO37	OUTPUTXBAR2	SPIA_PTE	I2CA_SCL	SCIA_TX		UARTA_TX	EQEP1_B			SYNCOUT		TDO	
GPIO39							EQEP1_INDEX			SYNCOUT	EQEP1_INDEX		
GPIO40				MCPWM1_2B			SCIB_TX	EQEP1_A					
GPIO41	MCPWM3_1A	SPIA_CLK		MCPWM1_2A			SCIB_RX	EQEP1_B					
GPIO43			OUTPUTXBAR6		I2CA_SCL	UARTA_TX		EQEP1_INDEX					
GPIO45			OUTPUTXBAR8		SPIA_POCI								
GPIO46													
GPIO224				OUTPUTXBAR3	SPIA_PICO		MCPWM1_1A		EQEP1_A		UARTA_TX		
GPIO226					SPIA_CLK		MCPWM1_1B		EQEP1_STROBE		UARTA_RX		
GPIO227	I2CA_SCL		MCPWM1_3A	OUTPUTXBAR1	MCPWM1_2B								
GPIO228			ADCSOCAO		SPIA_POCI		MCPWM1_2B		EQEP1_B				
GPIO230	I2CA_SDA		MCPWM1_3B		MCPWM1_2A	I2CA_SDA							
GPIO242			MCPWM1_2A	OUTPUTXBAR2	SPIA_PTE				EQEP1_INDEX				
GPIO243	XCLKOUT												
AIO225													
AIO231													
AIO232													
AIO233													
AIO237													
AIO238													
AIO239													

Table 8-6. GPIO Muxed Pins (continued)

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
AIO241													
AIO244													
AIO245													

8.7.2 Peripheral Muxing

For example, multiplexing for the GPIO6 pin is controlled by writing to GPAGMUX[13:12] and GPAMUX[13:12]. By writing to these bits, GPIO6 is configured as either a general-purpose digital I/O or one of several different peripheral functions. An example of GPyGMUX and GPyMUX selection and options for a single GPIO are shown in Table 8-7.

Note

The following table is for example only. Refer to the device data sheet to check the availability of GPIO6 on this device. If GPIO6 is available, the functions mentioned in the table may not match the actual functions available. See Section 8.7.1 for correct list of GPIOs and corresponding mux options for this device.

Table 8-7. GPIO and Peripheral Muxing

GPAGMUX1[13:12]	GPAMUX1[13:12]	Pin Functionality
00	00	GPIO6
00	01	Peripheral 1
00	10	Peripheral 2
00	11	Peripheral 3
01	00	GPIO6
01	01	Peripheral 4
01	10	Peripheral 5
01	11	
10	00	GPIO6
10	01	
10	10	Peripheral 6
10	11	Peripheral 7
11	00	GPIO6
11	01	Peripheral 8
11	10	Peripheral 9
11	11	Peripheral 10

The devices have different multiplexing schemes. If a peripheral is not available on a particular device, that mux selection is reserved on that device and must not be used.

CAUTION

If a reserved GPIO mux configuration that is not mapped to either a peripheral or GPIO mode is selected, the state of the pin is undefined and the pin is driven. Unimplemented configurations are for future expansion and must not be selected. In the device mux table (see the data sheet), these options are indicated as Reserved or left blank.

Some peripherals can be assigned to more than one pin by way of the mux registers. For example, OUTPUTXBAR1 can be assigned to GPIOs p, q, or r (where p, q, and r are example GPIO numbers), depending on individual system requirements. An example of this is shown in Table 8-8.

Note

The following table is for example only. Bit ranges cannot correspond to OUTPUTXBAR1 on this device. See Section 8.7.1 for correct list of GPIOs and corresponding mux options for this device.

If none or more then one of the GPIO pins is configured as peripheral input pins, then that GPIO is set to a hard-wired default value.

Table 8-8. Peripheral Muxing (Multiple Pins Assigned)

GMUX Configuration	MUX Configuration	
Choice 1: GPIOp	GPyGMUX1[5:4] = 01	GPyMUX1[5:4] = 01
or Choice 2: GPIOq	GPyGMUX2[17:16] = 00	GPyMUX2[17:16] = 01
or Choice 3: GPIOr	GPyGMUX1[7:6] = 01	GPyMUX1[7:6] = 01

8.8 Internal Pullup Configuration Requirements

On reset, GPIOs are in input mode and have the internal pullups disabled. An undriven input can float to a mid-rail voltage and cause wasted shoot-through current on the input buffer. The user must always put each GPIO in one of these configurations:

- Input mode and driven on the board by another component to a level above V_{ih} or below V_{il}
- Input mode with GPIO internal pullup enabled
- Output mode

On devices with lesser pin count packages, pull-ups on unbonded GPIOs are by default enabled to prevent floating inputs. The user must take care to avoid disabling these pullups in the application code.

On devices with larger pin count packages, the pullups for any internally unbonded GPIO must be enabled to prevent floating inputs. TI has provided functions in the controlSUITE/C2000Ware that users can call to enable the pullup on any unbonded GPIO for the package in use. This function, GPIO_EnabledUnbondedIOPullups(), resides in the (Device)_Sysctrl.c file and is called by default from InitSysCtrl(). The user must take care to avoid disabling these pullups in the application code.

8.9 Open-Drain Configuration Requirements

To configure GPIO as an open-drain pin, write 0 to the GPxDAT register for the pin followed by writing 0 to the GPxDIR register. When writing 0 to the GPxDIR register, the pin is floating; when writing 1 to the GPxDIR register, the pin is driven low.

8.10 Software

8.10.1 GPIO Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/gpio

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

8.10.1.1 Device GPIO Setup

FILE: gpio_ex1_setup.c

Configures the device GPIO into two different configurations This code is verbose to illustrate how the GPIO could be setup. In a real application, lines of code can be combined for improved code size and efficiency.

This example only sets-up the GPIO. Nothing is actually done with the pins after setup.

In general:

- All pullup resistors are enabled. For PWMs this may not be desired.
- Input qual for communication ports (SPI, SCI, I2C) is asynchronous
- Input qual for Trip pins (TZ) is asynchronous
- Input qual for eCAP and eQEP signals is synch to SYSCLKOUT
- Input qual for some I/O's and __interrupts may have a sampling window

8.10.1.2 Device GPIO Toggle

FILE: gpio_ex2_toggle.c

Configures the device GPIO through the sysconfig file. The GPIO pin is toggled in the infinite loop. In order to migrate the project within syscfg to any device, click the swtich button under the device view and select your corresponding device to migrate, saving the project will auto-migrate your project settings.

: This example project has support for migration across our C2000 device families. If you are wanting to build this project from launchpad or controlCARD, please specify in the .syscfg file the board you're using. At any time you can select another device to migrate this example.

8.10.1.3 Device GPIO Interrupt

FILE: gpio_ex3_interrupt.c

Configures the device GPIOs through the sysconfig file. One GPIO output pin, and one GPIO input pin is configured. The example then configures the GPIO input pin to be the source of an external interrupt which toggles the GPIO output pin.

8.10.1.4 External Interrupt (XINT)

FILE: gpio_ex4_aio_external_interrupt.c

In this example AIO pins are configured as digital inputs. Two other GPIO signals (connected externally to AIO pins) are toggled in software to trigger external interrupt through AIO227 and AIO230 (AIO227 assigned to XINT1 and AIO230 assigned to XINT2). The user is required to externally connect these signals for the program to work properly. Each interrupt is fired in sequence: XINT1 first and then XINT2.

- GPIO5 will go high outside of the interrupts and low within the interrupts. This signal can be monitored on a scope. *External Connections*
- Connect GPIO0 to AIO227. AIO227 will be assigned to XINT1
- Connect GPIO1 to AIO230. AIO230 will be assigned to XINT2
- GPIO5 can be monitored on an oscilloscope

Watch Variables
- xint1Count for the number of times through XINT1 interrupt
- xint2Count for the number of times through XINT2 interrupt
- loopCount for the number of times through the idle loop

8.10.2 LED Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/led

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

8.11 GPIO Registers

This Section describes the GPIO Registers.

8.11.1 GPIO Base Address Table

Table 8-9. GPIO Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
GpioCtrlRegs	GPIO_CTRL_REGS	GPIOCTRL_BASE	0x0000_7C00	-	YES
GpioDataRegs	GPIO_DATA_REGS	GPIODATA_BASE	0x0000_7F00	-	YES
GpioDataReadRegs	GPIO_DATA_READ_REGS	GPIODATAREAD_BASE	0x0000_7F80	-	YES

8.11.2 GPIO_CTRL_REGS Registers

Table 8-10 lists the memory-mapped registers for the GPIO_CTRL_REGS registers. All register offset addresses not listed in Table 8-10 should be considered as reserved locations and the register contents should not be modified.

Table 8-10. GPIO_CTRL_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	GPACTRL	GPIO A Qualification Sampling Period Control (GPIO0 to 31)	EALLOW
2h	GPAQSEL1	GPIO A Qualifier Select 1 Register (GPIO0 to 15)	EALLOW
4h	GPAQSEL2	GPIO A Qualifier Select 2 Register (GPIO16 to 31)	EALLOW
6h	GPAMUX1	GPIO A Mux 1 Register (GPIO0 to 15)	EALLOW
8h	GPAMUX2	GPIO A Mux 2 Register (GPIO16 to 31)	EALLOW
Ah	GPADIR	GPIO A Direction Register (GPIO0 to 31)	EALLOW
Ch	GPAPUD	GPIO A Pull Up Disable Register (GPIO0 to 31)	EALLOW
10h	GPAINV	GPIO A Input Polarity Invert Registers (GPIO0 to 31)	EALLOW
14h	GPAAMSEL	GPIO A Analog Mode Select register (GPIO0 to GPIO31)	EALLOW
20h	GPAGMUX1	GPIO A Peripheral Group Mux (GPIO0 to 15)	EALLOW
22h	GPAGMUX2	GPIO A Peripheral Group Mux (GPIO16 to 31)	EALLOW
3Ch	GPALOCK	GPIO A Lock Configuration Register (GPIO0 to 31)	EALLOW
3Eh	GPACR	GPIO A Lock Commit Register (GPIO0 to 31)	EALLOW
40h	GPBCTRL	GPIO B Qualification Sampling Period Control (GPIO32 to 63)	EALLOW
42h	GPBQSEL1	GPIO B Qualifier Select 1 Register (GPIO32 to 47)	EALLOW
44h	GPBQSEL2	GPIO B Qualifier Select 2 Register (GPIO48 to 63)	EALLOW
46h	GPBMUX1	GPIO B Mux 1 Register (GPIO32 to 47)	EALLOW
48h	GPBMUX2	GPIO B Mux 2 Register (GPIO48 to 63)	EALLOW
4Ah	GPBDIR	GPIO B Direction Register (GPIO32 to 63)	EALLOW
4Ch	GPBPUD	GPIO B Pull Up Disable Register (GPIO32 to 63)	EALLOW
50h	GPBINV	GPIO B Input Polarity Invert Registers (GPIO32 to 63)	EALLOW
60h	GPBGMUX1	GPIO B Peripheral Group Mux (GPIO32 to 47)	EALLOW
62h	GPBGMUX2	GPIO B Peripheral Group Mux (GPIO48 to 63)	EALLOW
7Ch	GPBLOCK	GPIO B Lock Configuration Register (GPIO32 to 63)	EALLOW
7Eh	GPBCR	GPIO B Lock Commit Register (GPIO32 to 63)	EALLOW
1C0h	GPHCTRL	GPIO H Qualification Sampling Period Control (GPIO224 to 255)	EALLOW
1C2h	GPHQSEL1	GPIO H Qualifier Select 1 Register (GPIO224 to 239)	EALLOW
1C4h	GPHQSEL2	GPIO H Qualifier Select 2 Register (GPIO240 to 255)	EALLOW
1C6h	GPHMUX1	GPIO H Mux 1 Register (GPIO224 to 239)	EALLOW
1C8h	GPHMUX2	GPIO H Mux 2 Register (GPIO240 to 255)	EALLOW
1CAh	GPHDIR	GPIO H Direction Register (GPIO224 to 255)	EALLOW
1CCh	GPHPUD	GPIO H Pull Up Disable Register (GPIO224 to 255)	EALLOW
1D0h	GPHINV	GPIO H Input Polarity Invert Registers (GPIO224 to 255)	EALLOW
1D4h	GPHAMSEL	GPIO H Analog Mode Select register (GPIO224 to GPIO255)	EALLOW
1E0h	GPHGMUX1	GPIO H Peripheral Group Mux (GPIO224 to 239)	EALLOW
1E2h	GPHGMUX2	GPIO H Peripheral Group Mux (GPIO240 to 255)	EALLOW
1FCh	GPHLOCK	GPIO H Lock Configuration Register (GPIO224 to 255)	EALLOW

Table 8-10. GPIO_CTRL_REGS Registers (continued)

Offset	Acronym	Register Name	Write Protection
1FEh	GPHCR	GPIO H Lock Commit Register (GPIO224 to 255)	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 8-11 shows the codes that are used for access types in this section.

Table 8-11. GPIO_CTRL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

8.11.2.1 GPACTRL Register (Offset = 0h) [Reset = 00000000h]

GPACTRL is shown in Figure 8-4 and described in Table 8-12.

Return to the Summary Table.

GPIO A Qualification Sampling Period Control (GPIO0 to 31)

Figure 8-4. GPACTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUALPRD3								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 8-12. GPACTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	QUALPRD3	R/W	0h	Qualification sampling period for GPIO24 to GPIO31: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
23-16	QUALPRD2	R/W	0h	Qualification sampling period for GPIO16 to GPIO23: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
15-8	QUALPRD1	R/W	0h	Qualification sampling period for GPIO8 to GPIO15: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	Qualification sampling period for GPIO0 to GPIO7: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

8.11.2.2 GPAQSEL1 Register (Offset = 2h) [Reset = 00000000h]

GPAQSEL1 is shown in Figure 8-5 and described in Table 8-13.

Return to the Summary Table.

GPIO A Qualifier Select 1 Register (GPIO0 to 15)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 8-5. GPAQSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO13		GPIO12		GPIO11		GPIO10		GPIO9		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-13. GPAQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	GPIO13	R/W	0h	Select input qualification type for GPIO13: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO12	R/W	0h	Select input qualification type for GPIO12: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
23-22	GPIO11	R/W	0h	Select input qualification type for GPIO11: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	GPIO10	R/W	0h	Select input qualification type for GPIO10: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
19-18	GPIO9	R/W	0h	Select input qualification type for GPIO9: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	RESERVED	R/W	0h	Reserved

Table 8-13. GPAQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	GPIO7	R/W	0h	Select input qualification type for GPIO7: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO6	R/W	0h	Select input qualification type for GPIO6: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	GPIO5	R/W	0h	Select input qualification type for GPIO5: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO4	R/W	0h	Select input qualification type for GPIO4: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO3	R/W	0h	Select input qualification type for GPIO3: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO2	R/W	0h	Select input qualification type for GPIO2: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO1	R/W	0h	Select input qualification type for GPIO1: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO0	R/W	0h	Select input qualification type for GPIO0: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

8.11.2.3 GPAQSEL2 Register (Offset = 4h) [Reset = 00000000h]

GPAQSEL2 is shown in Figure 8-6 and described in Table 8-14.

Return to the Summary Table.

GPIO A Qualifier Select 2 Register (GPIO16 to 31)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 8-6. GPAQSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		GPIO30		GPIO29		GPIO28		RESERVED		RESERVED		RESERVED		GPIO24	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO23		RESERVED		RESERVED		RESERVED		GPIO19		GPIO18		RESERVED		GPIO16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-14. GPAQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	GPIO30	R/W	0h	Select input qualification type for GPIO30: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO29	R/W	0h	Select input qualification type for GPIO29: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	GPIO28	R/W	0h	Select input qualification type for GPIO28: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	GPIO24	R/W	0h	Select input qualification type for GPIO24: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO23	R/W	0h	Select input qualification type for GPIO23: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 8-14. GPAQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	GPIO19	R/W	0h	Select input qualification type for GPIO19: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO18	R/W	0h	Select input qualification type for GPIO18: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	RESERVED	R/W	0h	Reserved
1-0	GPIO16	R/W	0h	Select input qualification type for GPIO16: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

8.11.2.4 GPAMUX1 Register (Offset = 6h) [Reset = 00000000h]

GPAMUX1 is shown in Figure 8-7 and described in Table 8-15.

Return to the Summary Table.

GPIO A Mux 1 Register (GPIO0 to 15)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 8-7. GPAMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO13		GPIO12		GPIO11		GPIO10		GPIO9		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-15. GPAMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	GPIO13	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO12	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO11	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO10	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO9	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	RESERVED	R/W	0h	Reserved
15-14	GPIO7	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO6	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO5	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO4	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO3	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO2	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO1	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO0	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

8.11.2.5 GPAMUX2 Register (Offset = 8h) [Reset = 00000000h]

GPAMUX2 is shown in Figure 8-8 and described in Table 8-16.

Return to the Summary Table.

GPIO A Mux 2 Register (GPIO16 to 31)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 8-8. GPAMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		GPIO30		GPIO29		GPIO28		RESERVED		RESERVED		RESERVED		GPIO24	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO23		RESERVED		RESERVED		RESERVED		GPIO19		GPIO18		RESERVED		GPIO16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-16. GPAMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	GPIO30	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO29	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO28	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	GPIO24	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO23	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	GPIO19	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO18	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	RESERVED	R/W	0h	Reserved
1-0	GPIO16	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

8.11.2.6 GPADIR Register (Offset = Ah) [Reset = 00000000h]

GPADIR is shown in Figure 8-9 and described in Table 8-17.

Return to the Summary Table.

GPIO A Direction Register (GPIO0 to 31)
Controls direction of GPIO pins when the specified pin is configured in GPIO mode.
0: Configures pin as input.
1: Configures pin as output.
Reading the register returns the current value of the register setting.

Figure 8-9. GPADIR Register

31	30	29	28	27	26	25	24
RESERVED	GPIO30	GPIO29	GPIO28	RESERVED	RESERVED	RESERVED	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	RESERVED	RESERVED	RESERVED	GPIO19	GPIO18	RESERVED	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-17. GPADIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	GPIO30	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
29	GPIO29	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
28	GPIO28	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	GPIO24	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
23	GPIO23	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	GPIO19	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
18	GPIO18	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
17	RESERVED	R/W	0h	Reserved

Table 8-17. GPADIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	GPIO16	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	GPIO13	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
12	GPIO12	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
11	GPIO11	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
10	GPIO10	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
9	GPIO9	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
8	RESERVED	R/W	0h	Reserved
7	GPIO7	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
6	GPIO6	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
5	GPIO5	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
4	GPIO4	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
3	GPIO3	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
2	GPIO2	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
1	GPIO1	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
0	GPIO0	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn

8.11.2.7 GPAPUD Register (Offset = Ch) [Reset = FFFFFFFFh]

GPAPUD is shown in Figure 8-10 and described in Table 8-18.

Return to the Summary Table.

GPIO A Pull Up Disable Register (GPIO0 to 31)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 8-10. GPAPUD Register

31	30	29	28	27	26	25	24
RESERVED	GPIO30	GPIO29	GPIO28	RESERVED	RESERVED	RESERVED	GPIO24
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
GPIO23	RESERVED	RESERVED	RESERVED	GPIO19	GPIO18	RESERVED	GPIO16
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 8-18. GPAPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	1h	Reserved
30	GPIO30	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
29	GPIO29	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
28	GPIO28	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
27	RESERVED	R/W	1h	Reserved
26	RESERVED	R/W	1h	Reserved
25	RESERVED	R/W	1h	Reserved
24	GPIO24	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
23	GPIO23	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
22	RESERVED	R/W	1h	Reserved
21	RESERVED	R/W	1h	Reserved
20	RESERVED	R/W	1h	Reserved
19	GPIO19	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
18	GPIO18	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

Table 8-18. GPAPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	RESERVED	R/W	1h	Reserved
16	GPIO16	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
15	RESERVED	R/W	1h	Reserved
14	RESERVED	R/W	1h	Reserved
13	GPIO13	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
12	GPIO12	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
11	GPIO11	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
10	GPIO10	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
9	GPIO9	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
8	RESERVED	R/W	1h	Reserved
7	GPIO7	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
6	GPIO6	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
5	GPIO5	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
4	GPIO4	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
3	GPIO3	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
2	GPIO2	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
1	GPIO1	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
0	GPIO0	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

8.11.2.8 GPAINV Register (Offset = 10h) [Reset = 00000000h]

GPAINV is shown in Figure 8-11 and described in Table 8-19.

Return to the Summary Table.

GPIO A Input Polarity Invert Registers (GPIO0 to 31)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 8-11. GPAINV Register

31	30	29	28	27	26	25	24
RESERVED	GPIO30	GPIO29	GPIO28	RESERVED	RESERVED	RESERVED	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	RESERVED	RESERVED	RESERVED	GPIO19	GPIO18	RESERVED	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-19. GPAINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	GPIO30	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
29	GPIO29	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
28	GPIO28	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	GPIO24	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
23	GPIO23	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	GPIO19	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
18	GPIO18	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
17	RESERVED	R/W	0h	Reserved

Table 8-19. GPAINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	GPIO16	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	GPIO13	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
12	GPIO12	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
11	GPIO11	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
10	GPIO10	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
9	GPIO9	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
8	RESERVED	R/W	0h	Reserved
7	GPIO7	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
6	GPIO6	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
5	GPIO5	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
4	GPIO4	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
3	GPIO3	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
2	GPIO2	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
1	GPIO1	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
0	GPIO0	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

8.11.2.9 GPAAMSEL Register (Offset = 14h) [Reset = FFFFFFFFh]

GPAAMSEL is shown in Figure 8-12 and described in Table 8-20.

Return to the Summary Table.

GPIO A Analog Mode Select register (GPIO0 to GPIO31)
Selects between digital and analog functionality for GPIO pins.
0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers
1: The analog function of the pin is enabled and the pin is capable of analog functions
Reading the register returns the current value of the register setting.
Note:
[1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect.

Figure 8-12. GPAAMSEL Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	GPIO28	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 8-20. GPAAMSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	1h	Reserved
30	RESERVED	R/W	1h	Reserved
29	RESERVED	R/W	1h	Reserved
28	GPIO28	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
27	RESERVED	R/W	1h	Reserved
26	RESERVED	R/W	1h	Reserved
25	RESERVED	R/W	1h	Reserved
24	RESERVED	R/W	1h	Reserved
23	RESERVED	R/W	1h	Reserved
22	RESERVED	R/W	1h	Reserved
21	RESERVED	R/W	1h	Reserved
20	RESERVED	R/W	1h	Reserved
19	RESERVED	R/W	1h	Reserved
18	RESERVED	R/W	1h	Reserved
17	RESERVED	R/W	1h	Reserved
16	RESERVED	R/W	1h	Reserved
15	RESERVED	R/W	1h	Reserved

Table 8-20. GPAAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	RESERVED	R/W	1h	Reserved
13	GPIO13	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
12	GPIO12	R/W	1h	Analog Mode select for this pin Reset type: SYSRSn
11	RESERVED	R/W	1h	Reserved
10	RESERVED	R/W	1h	Reserved
9	RESERVED	R/W	1h	Reserved
8	RESERVED	R/W	1h	Reserved
7	RESERVED	R/W	1h	Reserved
6	RESERVED	R/W	1h	Reserved
5	RESERVED	R/W	1h	Reserved
4	RESERVED	R/W	1h	Reserved
3	RESERVED	R/W	1h	Reserved
2	RESERVED	R/W	1h	Reserved
1	RESERVED	R/W	1h	Reserved
0	RESERVED	R/W	1h	Reserved

8.11.2.10 GPAGMUX1 Register (Offset = 20h) [Reset = 00000000h]

GPAGMUX1 is shown in Figure 8-13 and described in Table 8-21.

Return to the Summary Table.

GPIO A Peripheral Group Mux (GPIO0 to 15)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 8-13. GPAGMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO13		GPIO12		GPIO11		GPIO10		GPIO9		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-21. GPAGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	GPIO13	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO12	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	GPIO11	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	GPIO10	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
19-18	GPIO9	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	RESERVED	R/W	0h	Reserved
15-14	GPIO7	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	GPIO6	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	GPIO5	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	GPIO4	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO3	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO2	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	GPIO1	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO0	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

8.11.2.11 GPAGMUX2 Register (Offset = 22h) [Reset = 00000000h]

GPAGMUX2 is shown in Figure 8-14 and described in Table 8-22.

Return to the Summary Table.

GPIO A Peripheral Group Mux (GPIO16 to 31)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 8-14. GPAGMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		GPIO30		GPIO29		GPIO28		RESERVED		RESERVED		RESERVED		GPIO24	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO23		RESERVED		RESERVED		RESERVED		GPIO19		GPIO18		RESERVED		GPIO16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-22. GPAGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	GPIO30	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
27-26	GPIO29	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	GPIO28	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	GPIO24	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO23	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	GPIO19	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO18	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	RESERVED	R/W	0h	Reserved
1-0	GPIO16	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

8.11.2.12 GPALOCK Register (Offset = 3Ch) [Reset = 00000000h]

GPALOCK is shown in Figure 8-15 and described in Table 8-23.

Return to the Summary Table.

GPIO A Lock Configuration Register (GPIO0 to 31)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 8-15. GPALOCK Register

31	30	29	28	27	26	25	24
RESERVED	GPIO30	GPIO29	GPIO28	RESERVED	RESERVED	RESERVED	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	RESERVED	RESERVED	RESERVED	GPIO19	GPIO18	RESERVED	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-23. GPALOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	GPIO30	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
29	GPIO29	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
28	GPIO28	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	GPIO24	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
23	GPIO23	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	GPIO19	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
18	GPIO18	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
17	RESERVED	R/W	0h	Reserved

Table 8-23. GPALOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	GPIO16	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	GPIO13	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
12	GPIO12	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
11	GPIO11	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
10	GPIO10	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
9	GPIO9	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
8	RESERVED	R/W	0h	Reserved
7	GPIO7	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
6	GPIO6	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
5	GPIO5	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
4	GPIO4	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
3	GPIO3	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
2	GPIO2	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
1	GPIO1	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
0	GPIO0	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

8.11.2.13 GPACR Register (Offset = 3Eh) [Reset = 00000000h]

GPACR is shown in Figure 8-16 and described in Table 8-24.

Return to the Summary Table.

GPIO A Lock Commit Register (GPIO0 to 31)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 8-16. GPACR Register

31	30	29	28	27	26	25	24
RESERVED	GPIO30	GPIO29	GPIO28	RESERVED	RESERVED	RESERVED	GPIO24
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
GPIO23	RESERVED	RESERVED	RESERVED	GPIO19	GPIO18	RESERVED	GPIO16
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 8-24. GPACR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/WSonce	0h	Reserved
30	GPIO30	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
29	GPIO29	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
28	GPIO28	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
27	RESERVED	R/WSonce	0h	Reserved
26	RESERVED	R/WSonce	0h	Reserved
25	RESERVED	R/WSonce	0h	Reserved
24	GPIO24	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
23	GPIO23	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
22	RESERVED	R/WSonce	0h	Reserved
21	RESERVED	R/WSonce	0h	Reserved
20	RESERVED	R/WSonce	0h	Reserved
19	GPIO19	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
18	GPIO18	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
17	RESERVED	R/WSonce	0h	Reserved
16	GPIO16	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

Table 8-24. GPACR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RESERVED	R/WSonce	0h	Reserved
14	RESERVED	R/WSonce	0h	Reserved
13	GPIO13	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
12	GPIO12	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
11	GPIO11	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
10	GPIO10	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
9	GPIO9	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
8	RESERVED	R/WSonce	0h	Reserved
7	GPIO7	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
6	GPIO6	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
5	GPIO5	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
4	GPIO4	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
3	GPIO3	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
2	GPIO2	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
1	GPIO1	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
0	GPIO0	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

8.11.2.14 GPBCTRL Register (Offset = 40h) [Reset = 00000000h]

GPBCTRL is shown in Figure 8-17 and described in Table 8-25.

Return to the Summary Table.

GPIO B Qualification Sampling Period Control (GPIO32 to 63)

Figure 8-17. GPBCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 8-25. GPBCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	Reserved
23-16	RESERVED	R/W	0h	Reserved
15-8	QUALPRD1	R/W	0h	Qualification sampling period for GPIO40 to GPIO47: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	Qualification sampling period for GPIO32 to GPIO39: 0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

8.11.2.15 GPBQSEL1 Register (Offset = 42h) [Reset = 00000CC0h]

GPBQSEL1 is shown in Figure 8-18 and described in Table 8-26.

Return to the Summary Table.

GPIO B Qualifier Select 1 Register (GPIO32 to 47)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 8-18. GPBQSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO45		RESERVED		GPIO43		RESERVED		GPIO41		GPIO40	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO39		RESERVED		GPIO37		RESERVED		GPIO35		RESERVED		GPIO33		GPIO32	
R/W-0h		R/W-0h		R/W-3h		R/W-0h		R/W-3h		R/W-0h		R/W-0h		R/W-0h	

Table 8-26. GPBQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	GPIO45	R/W	0h	Select input qualification type for GPIO45: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	RESERVED	R/W	0h	Reserved
23-22	GPIO43	R/W	0h	Select input qualification type for GPIO43: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO41	R/W	0h	Select input qualification type for GPIO41: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
17-16	GPIO40	R/W	0h	Select input qualification type for GPIO40: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO39	R/W	0h	Select input qualification type for GPIO39: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 8-26. GPBQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	RESERVED	R/W	0h	Reserved
11-10	GPIO37	R/W	3h	Select input qualification type for GPIO37: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	RESERVED	R/W	0h	Reserved
7-6	GPIO35	R/W	3h	Select input qualification type for GPIO35: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	RESERVED	R/W	0h	Reserved
3-2	GPIO33	R/W	0h	Select input qualification type for GPIO33: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO32	R/W	0h	Select input qualification type for GPIO32: 0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

8.11.2.16 GPBQSEL2 Register (Offset = 44h) [Reset = 00000000h]

GPBQSEL2 is shown in Figure 8-19 and described in Table 8-27.

Return to the Summary Table.

GPIO B Qualifier Select 2 Register (GPIO48 to 63)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 8-19. GPBQSEL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-27. GPBQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

8.11.2.17 GPBMUX1 Register (Offset = 46h) [Reset = 00000CC0h]

GPBMUX1 is shown in Figure 8-20 and described in Table 8-28.

Return to the Summary Table.

GPIO B Mux 1 Register (GPIO32 to 47)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 8-20. GPBMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO45		RESERVED		GPIO43		RESERVED		GPIO41		GPIO40	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO39		RESERVED		GPIO37		RESERVED		GPIO35		RESERVED		GPIO33		GPIO32	
R/W-0h		R/W-0h		R/W-3h		R/W-0h		R/W-3h		R/W-0h		R/W-0h		R/W-0h	

Table 8-28. GPBMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	GPIO45	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	RESERVED	R/W	0h	Reserved
23-22	GPIO43	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO41	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO40	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO39	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	RESERVED	R/W	0h	Reserved
11-10	GPIO37	R/W	3h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	RESERVED	R/W	0h	Reserved
7-6	GPIO35	R/W	3h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	RESERVED	R/W	0h	Reserved
3-2	GPIO33	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO32	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

8.11.2.18 GPBMUX2 Register (Offset = 48h) [Reset = 00000000h]

GPBMUX2 is shown in Figure 8-21 and described in Table 8-29.

Return to the Summary Table.

GPIO B Mux 2 Register (GPIO48 to 63)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 8-21. GPBMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-29. GPBMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

8.11.2.19 GPBDIR Register (Offset = 4Ah) [Reset = 00000000h]

GPBDIR is shown in Figure 8-22 and described in Table 8-30.

Return to the Summary Table.

GPIO B Direction Register (GPIO32 to 63)
Controls direction of GPIO pins when the specified pin is configured in GPIO mode.
0: Configures pin as input.
1: Configures pin as output.
Reading the register returns the current value of the register setting.

Figure 8-22. GPBDIR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO45	RESERVED	GPIO43	RESERVED	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	RESERVED	GPIO37	RESERVED	GPIO35	RESERVED	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-30. GPBDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	GPIO45	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved

Table 8-30. GPBDIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	GPIO43	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
10	RESERVED	R/W	0h	Reserved
9	GPIO41	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
8	GPIO40	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
7	GPIO39	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
6	RESERVED	R/W	0h	Reserved
5	GPIO37	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
4	RESERVED	R/W	0h	Reserved
3	GPIO35	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
2	RESERVED	R/W	0h	Reserved
1	GPIO33	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
0	GPIO32	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn

8.11.2.20 GPBPUD Register (Offset = 4Ch) [Reset = FFFFFFFFh]

GPBPUD is shown in Figure 8-23 and described in Table 8-31.

Return to the Summary Table.

GPIO B Pull Up Disable Register (GPIO32 to 63)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 8-23. GPBPUD Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO45	RESERVED	GPIO43	RESERVED	GPIO41	GPIO40
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO39	RESERVED	GPIO37	RESERVED	GPIO35	RESERVED	GPIO33	GPIO32
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 8-31. GPBPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	1h	Reserved
30	RESERVED	R/W	1h	Reserved
29	RESERVED	R/W	1h	Reserved
28	RESERVED	R/W	1h	Reserved
27	RESERVED	R/W	1h	Reserved
26	RESERVED	R/W	1h	Reserved
25	RESERVED	R/W	1h	Reserved
24	RESERVED	R/W	1h	Reserved
23	RESERVED	R/W	1h	Reserved
22	RESERVED	R/W	1h	Reserved
21	RESERVED	R/W	1h	Reserved
20	RESERVED	R/W	1h	Reserved
19	RESERVED	R/W	1h	Reserved
18	RESERVED	R/W	1h	Reserved
17	RESERVED	R/W	1h	Reserved
16	RESERVED	R/W	1h	Reserved
15	RESERVED	R/W	1h	Reserved
14	RESERVED	R/W	1h	Reserved
13	GPIO45	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

Table 8-31. GPBPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	RESERVED	R/W	1h	Reserved
11	GPIO43	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
10	RESERVED	R/W	1h	Reserved
9	GPIO41	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
8	GPIO40	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
7	GPIO39	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
6	RESERVED	R/W	1h	Reserved
5	GPIO37	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
4	RESERVED	R/W	1h	Reserved
3	GPIO35	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
2	RESERVED	R/W	1h	Reserved
1	GPIO33	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn
0	GPIO32	R/W	1h	Pull-Up Disable control for this pin Reset type: SYSRSn

8.11.2.21 GPBINV Register (Offset = 50h) [Reset = 00000000h]

GPBINV is shown in Figure 8-24 and described in Table 8-32.

Return to the Summary Table.

GPIO B Input Polarity Invert Registers (GPIO32 to 63)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 8-24. GPBINV Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO45	RESERVED	GPIO43	RESERVED	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	RESERVED	GPIO37	RESERVED	GPIO35	RESERVED	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-32. GPBINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	GPIO45	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved

Table 8-32. GPBINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	GPIO43	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
10	RESERVED	R/W	0h	Reserved
9	GPIO41	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
8	GPIO40	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
7	GPIO39	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
6	RESERVED	R/W	0h	Reserved
5	GPIO37	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
4	RESERVED	R/W	0h	Reserved
3	GPIO35	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
2	RESERVED	R/W	0h	Reserved
1	GPIO33	R/W	0h	Input inversion control for this pin Reset type: SYSRSn
0	GPIO32	R/W	0h	Input inversion control for this pin Reset type: SYSRSn

8.11.2.22 GPBGMUX1 Register (Offset = 60h) [Reset = 00000CC0h]

GPBGMUX1 is shown in Figure 8-25 and described in Table 8-33.

Return to the Summary Table.

GPIO B Peripheral Group Mux (GPIO32 to 47)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 8-25. GPBGMUX1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO45		RESERVED		GPIO43		RESERVED		GPIO41		GPIO40	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO39		RESERVED		GPIO37		RESERVED		GPIO35		RESERVED		GPIO33		GPIO32	
R/W-0h		R/W-0h		R/W-3h		R/W-0h		R/W-3h		R/W-0h		R/W-0h		R/W-0h	

Table 8-33. GPBGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	GPIO45	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
25-24	RESERVED	R/W	0h	Reserved
23-22	GPIO43	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO41	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
17-16	GPIO40	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
15-14	GPIO39	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
13-12	RESERVED	R/W	0h	Reserved
11-10	GPIO37	R/W	3h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
9-8	RESERVED	R/W	0h	Reserved
7-6	GPIO35	R/W	3h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	RESERVED	R/W	0h	Reserved
3-2	GPIO33	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
1-0	GPIO32	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

8.11.2.23 GPBGMUX2 Register (Offset = 62h) [Reset = 00000000h]

GPBGMUX2 is shown in Figure 8-26 and described in Table 8-34.

Return to the Summary Table.

GPIO B Peripheral Group Mux (GPIO48 to 63)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 8-26. GPBGMUX2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-34. GPBGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

8.11.2.24 GPBLOCK Register (Offset = 7Ch) [Reset = 00000000h]

GPBLOCK is shown in Figure 8-27 and described in Table 8-35.

Return to the Summary Table.

GPIO B Lock Configuration Register (GPIO32 to 63)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 8-27. GPBLOCK Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO45	RESERVED	GPIO43	RESERVED	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	RESERVED	GPIO37	RESERVED	GPIO35	RESERVED	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-35. GPBLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	GPIO45	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved

Table 8-35. GPBLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	GPIO43	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
10	RESERVED	R/W	0h	Reserved
9	GPIO41	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
8	GPIO40	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
7	GPIO39	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
6	RESERVED	R/W	0h	Reserved
5	GPIO37	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
4	RESERVED	R/W	0h	Reserved
3	GPIO35	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
2	RESERVED	R/W	0h	Reserved
1	GPIO33	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn
0	GPIO32	R/W	0h	Configuration Lock bit for this pin Reset type: SYSRSn

8.11.2.25 GPBCR Register (Offset = 7Eh) [Reset = 00000000h]

GPBCR is shown in Figure 8-28 and described in Table 8-36.

Return to the Summary Table.

GPIO B Lock Commit Register (GPIO32 to 63)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 8-28. GPBCR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO45	RESERVED	GPIO43	RESERVED	GPIO41	GPIO40
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO39	RESERVED	GPIO37	RESERVED	GPIO35	RESERVED	GPIO33	GPIO32
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 8-36. GPBCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/WSonce	0h	Reserved
30	RESERVED	R/WSonce	0h	Reserved
29	RESERVED	R/WSonce	0h	Reserved
28	RESERVED	R/WSonce	0h	Reserved
27	RESERVED	R/WSonce	0h	Reserved
26	RESERVED	R/WSonce	0h	Reserved
25	RESERVED	R/WSonce	0h	Reserved
24	RESERVED	R/WSonce	0h	Reserved
23	RESERVED	R/WSonce	0h	Reserved
22	RESERVED	R/WSonce	0h	Reserved
21	RESERVED	R/WSonce	0h	Reserved
20	RESERVED	R/WSonce	0h	Reserved
19	RESERVED	R/WSonce	0h	Reserved
18	RESERVED	R/WSonce	0h	Reserved
17	RESERVED	R/WSonce	0h	Reserved
16	RESERVED	R/WSonce	0h	Reserved
15	RESERVED	R/WSonce	0h	Reserved
14	RESERVED	R/WSonce	0h	Reserved
13	GPIO45	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
12	RESERVED	R/WSonce	0h	Reserved
11	GPIO43	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

Table 8-36. GPBCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RESERVED	R/WSonce	0h	Reserved
9	GPIO41	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
8	GPIO40	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
7	GPIO39	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
6	RESERVED	R/WSonce	0h	Reserved
5	GPIO37	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
4	RESERVED	R/WSonce	0h	Reserved
3	GPIO35	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
2	RESERVED	R/WSonce	0h	Reserved
1	GPIO33	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn
0	GPIO32	R/WSonce	0h	Configuration lock commit bit for this pin Reset type: SYSRSn

8.11.2.26 GPHCTRL Register (Offset = 1C0h) [Reset = 00000000h]

GPHCTRL is shown in Figure 8-29 and described in Table 8-37.

Return to the Summary Table.

GPIO H Qualification Sampling Period Control (GPIO224 to 255)

Figure 8-29. GPHCTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

Table 8-37. GPHCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	Reserved
23-16	QUALPRD2	R/W	0h	0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/512 Reset type: SYSRSn
15-8	QUALPRD1	R/W	0h	0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/511 Reset type: SYSRSn
7-0	QUALPRD0	R/W	0h	0x00,QUALPRDx = PLLSYSCLK 0x01,QUALPRDx = PLLSYSCLK/2 0x02,QUALPRDx = PLLSYSCLK/4 .... 0xFF,QUALPRDx = PLLSYSCLK/510 Reset type: SYSRSn

8.11.2.27 GPHQSEL1 Register (Offset = 1C2h) [Reset = 00000000h]

GPHQSEL1 is shown in Figure 8-30 and described in Table 8-38.

Return to the Summary Table.

GPIO H Qualifier Select 1 Register (GPIO224 to 239)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 8-30. GPHQSEL1 Register

31	30	29	28	27	26	25	24
GPIO239		GPIO238		GPIO237		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		GPIO233		GPIO232	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO231		GPIO230		RESERVED		GPIO228	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO227		GPIO226		GPIO225		GPIO224	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-38. GPHQSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	GPIO239	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
29-28	GPIO238	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
27-26	GPIO237	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	GPIO233	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 8-38. GPHQSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17-16	GPIO232	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
15-14	GPIO231	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
13-12	GPIO230	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
11-10	RESERVED	R/W	0h	Reserved
9-8	GPIO228	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
7-6	GPIO227	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO226	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO225	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	GPIO224	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

8.11.2.28 GPHQSEL2 Register (Offset = 1C4h) [Reset = 00000000h]

GPHQSEL2 is shown in Figure 8-31 and described in Table 8-39.

Return to the Summary Table.

GPIO H Qualifier Select 2 Register (GPIO240 to 255)
Input qualification type:
0,0 Sync
0,1 Qualification (3 samples)
1,0 Qualification (6 samples)
1,1 Async (no Sync or Qualification)

Figure 8-31. GPHQSEL2 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		GPIO245		GPIO244	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO243		GPIO242		GPIO241		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-39. GPHQSEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	GPIO245	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
9-8	GPIO244	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn

Table 8-39. GPHQSEL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	GPIO243	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
5-4	GPIO242	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
3-2	GPIO241	R/W	0h	0,0,Sync 0,1,Qualification (3 samples) 1,0,Qualification (6 samples) 1,1,Async (no Sync or Qualification) Reset type: SYSRSn
1-0	RESERVED	R/W	0h	Reserved

8.11.2.29 GPHMUX1 Register (Offset = 1C6h) [Reset = 00000000h]

GPHMUX1 is shown in Figure 8-32 and described in Table 8-40.

Return to the Summary Table.

GPIO H Mux 1 Register (GPIO224 to 239)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 8-32. GPHMUX1 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		GPIO230		RESERVED		GPIO228	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO227		GPIO226		RESERVED		GPIO224	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-40. GPHMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	GPIO230	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	RESERVED	R/W	0h	Reserved
9-8	GPIO228	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO227	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO226	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	RESERVED	R/W	0h	Reserved
1-0	GPIO224	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

8.11.2.30 GPHMUX2 Register (Offset = 1C8h) [Reset = 00000000h]

GPHMUX2 is shown in Figure 8-33 and described in Table 8-41.

Return to the Summary Table.

GPIO H Mux 2 Register (GPIO240 to 255)
Defines pin-muxing selection for GPIO.
Notes:
The respective GPyGMUXn.GPIOz must be configured prior to this register to avoid intermediate peripheral selects being mapped to the GPIO. Refer to GPIO chapter for more details.

Figure 8-33. GPHMUX2 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO243		GPIO242		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-41. GPHMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	GPIO243	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO242	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

8.11.2.31 GPHDIR Register (Offset = 1CAh) [Reset = 00000000h]

GPHDIR is shown in Figure 8-34 and described in Table 8-42.

Return to the Summary Table.

GPIO H Direction Register (GPIO224 to 255)
Controls direction of GPIO pins when the specified pin is configured in GPIO mode.
0: Configures pin as input.
1: Configures pin as output.
Reading the register returns the current value of the register setting.

Figure 8-34. GPHDIR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	GPIO243	GPIO242	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	RESERVED	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-42. GPHDIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	GPIO243	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
18	GPIO242	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved

Table 8-42. GPHDIR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	GPIO230	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
5	RESERVED	R/W	0h	Reserved
4	GPIO228	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
3	GPIO227	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
2	GPIO226	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn
1	RESERVED	R/W	0h	Reserved
0	GPIO224	R/W	0h	Defines direction for this pin in GPIO mode Reset type: SYSRSn

8.11.2.32 GPHPUD Register (Offset = 1CCh) [Reset = FFFFFFFFh]

GPHPUD is shown in Figure 8-35 and described in Table 8-43.

Return to the Summary Table.

GPIO H Pull Up Disable Register (GPIO224 to 255)
Disables the Pull-Up on GPIO.
0: Enables the Pull-Up.
1: Disables the Pull-Up.
Reading the register returns the current value of the register setting.
Note:
[1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low.

Figure 8-35. GPHPUD Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	RESERVED	RESERVED	RESERVED	GPIO233	GPIO232
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 8-43. GPHPUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	1h	Reserved
30	RESERVED	R/W	1h	Reserved
29	RESERVED	R/W	1h	Reserved
28	RESERVED	R/W	1h	Reserved
27	RESERVED	R/W	1h	Reserved
26	RESERVED	R/W	1h	Reserved
25	RESERVED	R/W	1h	Reserved
24	RESERVED	R/W	1h	Reserved
23	RESERVED	R/W	1h	Reserved
22	RESERVED	R/W	1h	Reserved
21	GPIO245	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn

Table 8-43. GPHPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO244	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
19	GPIO243	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
18	GPIO242	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
17	GPIO241	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
16	RESERVED	R/W	1h	Reserved
15	GPIO239	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
14	GPIO238	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn

Table 8-43. GPHPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	GPIO237	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
12	RESERVED	R/W	1h	Reserved
11	RESERVED	R/W	1h	Reserved
10	RESERVED	R/W	1h	Reserved
9	GPIO233	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
8	GPIO232	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
7	GPIO231	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
6	GPIO230	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
5	RESERVED	R/W	1h	Reserved
4	GPIO228	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn

Table 8-43. GPHPUD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	GPIO227	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
2	GPIO226	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
1	GPIO225	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn
0	GPIO224	R/W	1h	0: Enables the Pull-Up. 1: Disables the Pull-Up. Reading the register returns the current value of the register setting. Note: [1] The Pull-Ups on the GPIO pins are disabled asynchronously when IORSn signal is low. When coming out of reset, the pull-ups will remain disabled until the user enables them selectively in software by writing to this register. Reset type: SYSRSn

8.11.2.33 GPHINV Register (Offset = 1D0h) [Reset = 00000000h]

GPHINV is shown in Figure 8-36 and described in Table 8-44.

Return to the Summary Table.

GPIO H Input Polarity Invert Registers (GPIO224 to 255)
Selects between non-inverted and inverted GPIO input to the device.
0: selects non-inverted GPIO input
1: selects inverted GPIO input
Reading the register returns the current value of the register setting.

Figure 8-36. GPHINV Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	RESERVED	RESERVED	RESERVED	GPIO233	GPIO232
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-44. GPHINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	GPIO245	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
20	GPIO244	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn

Table 8-44. GPHINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO243	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
18	GPIO242	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
17	GPIO241	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
16	RESERVED	R/W	0h	Reserved
15	GPIO239	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
14	GPIO238	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
13	GPIO237	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	GPIO233	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
8	GPIO232	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn

Table 8-44. GPHINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	GPIO231	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
6	GPIO230	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
5	RESERVED	R/W	0h	Reserved
4	GPIO228	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
3	GPIO227	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
2	GPIO226	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
1	GPIO225	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn
0	GPIO224	R/W	0h	0: selects non-inverted GPIO input 1: selects inverted GPIO input Notes: [1] Reading the register returns the current value of the register setting. Reset type: SYSRSn

8.11.2.34 GPHAMSEL Register (Offset = 1D4h) [Reset = FFFFFFFFh]

GPHAMSEL is shown in Figure 8-37 and described in Table 8-45.

Return to the Summary Table.

GPIO H Analog Mode Select register (GPIO224 to GPIO255)
Selects between digital and analog functionality for GPIO pins.
0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers
1: The analog function of the pin is enabled and the pin is capable of analog functions
Reading the register returns the current value of the register setting.
Note:
[1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, t

Figure 8-37. GPHAMSEL Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	RESERVED
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	RESERVED	RESERVED	RESERVED	GPIO233	GPIO232
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 8-45. GPHAMSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	1h	Reserved
30	RESERVED	R/W	1h	Reserved
29	RESERVED	R/W	1h	Reserved
28	RESERVED	R/W	1h	Reserved
27	RESERVED	R/W	1h	Reserved
26	RESERVED	R/W	1h	Reserved
25	RESERVED	R/W	1h	Reserved
24	RESERVED	R/W	1h	Reserved
23	RESERVED	R/W	1h	Reserved
22	RESERVED	R/W	1h	Reserved
21	GPIO245	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

Table 8-45. GPHAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO244	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
19	GPIO243	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
18	GPIO242	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
17	GPIO241	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
16	RESERVED	R/W	1h	Reserved
15	GPIO239	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

Table 8-45. GPHAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	GPIO238	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
13	GPIO237	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
12	RESERVED	R/W	1h	Reserved
11	RESERVED	R/W	1h	Reserved
10	RESERVED	R/W	1h	Reserved
9	GPIO233	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
8	GPIO232	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
7	GPIO231	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

Table 8-45. GPHAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	GPIO230	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
5	RESERVED	R/W	1h	Reserved
4	GPIO228	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
3	GPIO227	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
2	GPIO226	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn
1	GPIO225	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

Table 8-45. GPHAMSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	GPIO224	R/W	1h	0: The analog function of the pin is disabled and the pin is capable of digital functions as specified by the other GPIO configuration registers 1: The analog function of the pin is enabled and the pin is capable of analog functions Reading the register returns the current value of the register setting. Note: [1] This register and bits are only valid for GPIO signals that share analog function through a unified I/O pad. For all the IOs, the corresponding bits in these registers dont have any affect. Reset type: SYSRSn

8.11.2.35 GPHGMUX1 Register (Offset = 1E0h) [Reset = 00000000h]

GPHGMUX1 is shown in Figure 8-38 and described in Table 8-46.

Return to the Summary Table.

GPIO H Peripheral Group Mux (GPIO224 to 239)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 8-38. GPHGMUX1 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		GPIO230		RESERVED		GPIO228	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO227		GPIO226		RESERVED		GPIO224	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-46. GPHGMUX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	GPIO230	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
11-10	RESERVED	R/W	0h	Reserved
9-8	GPIO228	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
7-6	GPIO227	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO226	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	RESERVED	R/W	0h	Reserved
1-0	GPIO224	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn

8.11.2.36 GPHGMUX2 Register (Offset = 1E2h) [Reset = 00000000h]

GPHGMUX2 is shown in Figure 8-39 and described in Table 8-47.

Return to the Summary Table.

GPIO H Peripheral Group Mux (GPIO240 to 255)
Defines pin-muxing selection for GPIO.
Notes:
[1]For complete pin-mux selection on GPIOx, GPAMUXy.GPIOx configuration is also required.

Figure 8-39. GPHGMUX2 Register

31	30	29	28	27	26	25	24
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO243		GPIO242		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 8-47. GPHGMUX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	GPIO243	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
5-4	GPIO242	R/W	0h	Defines pin-muxing selection for GPIO Reset type: SYSRSn
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

8.11.2.37 GPHLOCK Register (Offset = 1FCh) [Reset = 00000000h]

GPHLOCK is shown in Figure 8-40 and described in Table 8-48.

Return to the Summary Table.

GPIO H Lock Configuration Register (GPIO224 to 255)
GPIO Configuration Lock for GPIO.
0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed
1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin

Figure 8-40. GPHLOCK Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	RESERVED	RESERVED	RESERVED	GPIO233	GPIO232
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-48. GPHLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	GPIO245	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
20	GPIO244	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn

Table 8-48. GPHLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	GPIO243	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
18	GPIO242	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
17	GPIO241	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
16	RESERVED	R/W	0h	Reserved
15	GPIO239	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
14	GPIO238	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
13	GPIO237	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	GPIO233	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn

Table 8-48. GPHLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	GPIO232	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
7	GPIO231	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
6	GPIO230	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
5	RESERVED	R/W	0h	Reserved
4	GPIO228	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
3	GPIO227	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
2	GPIO226	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
1	GPIO225	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn
0	GPIO224	R/W	0h	1: Locks changes to the bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx registers which control the same pin 0: Bits in GPyMUX1, GPyMUX2, GPyDIR, GPyINV, GPyODR, GPyAMSEL, GPyGMUX1, GPyGMUX2 and GPyCSELx register which control the same pin can be changed Reset type: SYSRSn

8.11.2.38 GPHCR Register (Offset = 1FEh) [Reset = 00000000h]

GPHCR is shown in Figure 8-41 and described in Table 8-49.

Return to the Summary Table.

GPIO H Lock Commit Register (GPIO224 to 255)
GPIO Configuration Lock Commit for GPIO:
1: Locks changes to the bit in GPyLOCK register which controls the same pin
0: Bit in the GPyLOCK register which controls the same pin can be changed

Figure 8-41. GPHCR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	RESERVED
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	RESERVED	RESERVED	RESERVED	GPIO233	GPIO232
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 8-49. GPHCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/WSonce	0h	Reserved
30	RESERVED	R/WSonce	0h	Reserved
29	RESERVED	R/WSonce	0h	Reserved
28	RESERVED	R/WSonce	0h	Reserved
27	RESERVED	R/WSonce	0h	Reserved
26	RESERVED	R/WSonce	0h	Reserved
25	RESERVED	R/WSonce	0h	Reserved
24	RESERVED	R/WSonce	0h	Reserved
23	RESERVED	R/WSonce	0h	Reserved
22	RESERVED	R/WSonce	0h	Reserved
21	GPIO245	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
20	GPIO244	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
19	GPIO243	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn

Table 8-49. GPHCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	GPIO242	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
17	GPIO241	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
16	RESERVED	R/WSonce	0h	Reserved
15	GPIO239	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
14	GPIO238	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
13	GPIO237	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
12	RESERVED	R/WSonce	0h	Reserved
11	RESERVED	R/WSonce	0h	Reserved
10	RESERVED	R/WSonce	0h	Reserved
9	GPIO233	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
8	GPIO232	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
7	GPIO231	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
6	GPIO230	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
5	RESERVED	R/WSonce	0h	Reserved
4	GPIO228	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn

Table 8-49. GPHCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	GPIO227	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
2	GPIO226	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
1	GPIO225	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn
0	GPIO224	R/WSonce	0h	1: Locks changes to the bit in GPyLOCK register which controls the same pin 0: Bit in the GPyLOCK register which controls the same pin can be changed Reset type: SYSRSn

8.11.3 GPIO_DATA_REGS Registers

Table 8-50 lists the memory-mapped registers for the GPIO_DATA_REGS registers. All register offset addresses not listed in Table 8-50 should be considered as reserved locations and the register contents should not be modified.

Table 8-50. GPIO_DATA_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	GPADAT	GPIO A Data Register (GPIO0 to 31)	
2h	GPASET	GPIO A Data Set Register (GPIO0 to 31)	
4h	GPACLEAR	GPIO A Data Clear Register (GPIO0 to 31)	
6h	GPATOGGLE	GPIO A Data Toggle Register (GPIO0 to 31)	
8h	GPBDAT	GPIO B Data Register (GPIO32 to 63)	
Ah	GPBSET	GPIO B Data Set Register (GPIO32 to 63)	
Ch	GPBCLEAR	GPIO B Data Clear Register (GPIO32 to 63)	
Eh	GPBTOGGLE	GPIO B Data Toggle Register (GPIO32 to 63)	
38h	GPHDAT	GPIO H Data Register (GPIO224 to 255)	
3Ah	GPHSET	GPIO H Data Set Register (GPIO224 to 255)	
3Ch	GPHCLEAR	GPIO H Data Clear Register (GPIO224 to 255)	
3Eh	GPHTOGGLE	GPIO H Data Toggle Register (GPIO224 to 255)	

Complex bit access types are encoded to fit into small table cells. Table 8-51 shows the codes that are used for access types in this section.

Table 8-51. GPIO_DATA_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

8.11.3.1 GPADAT Register (Offset = 0h) [Reset = 00000000h]

GPADAT is shown in Figure 8-42 and described in Table 8-52.

Return to the Summary Table.

GPIO A Data Register (GPIO0 to 31)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register.

Figure 8-42. GPADAT Register

31	30	29	28	27	26	25	24
RESERVED	GPIO30	GPIO29	GPIO28	RESERVED	RESERVED	RESERVED	GPIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
GPIO23	RESERVED	RESERVED	RESERVED	GPIO19	GPIO18	RESERVED	GPIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-52. GPADAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	GPIO30	R/W	0h	Data Register for this pin Reset type: SYSRSn
29	GPIO29	R/W	0h	Data Register for this pin Reset type: SYSRSn
28	GPIO28	R/W	0h	Data Register for this pin Reset type: SYSRSn
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	GPIO24	R/W	0h	Data Register for this pin Reset type: SYSRSn
23	GPIO23	R/W	0h	Data Register for this pin Reset type: SYSRSn
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	GPIO19	R/W	0h	Data Register for this pin Reset type: SYSRSn

Table 8-52. GPADAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	GPIO18	R/W	0h	Data Register for this pin Reset type: SYSRSn
17	RESERVED	R/W	0h	Reserved
16	GPIO16	R/W	0h	Data Register for this pin Reset type: SYSRSn
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	GPIO13	R/W	0h	Data Register for this pin Reset type: SYSRSn
12	GPIO12	R/W	0h	Data Register for this pin Reset type: SYSRSn
11	GPIO11	R/W	0h	Data Register for this pin Reset type: SYSRSn
10	GPIO10	R/W	0h	Data Register for this pin Reset type: SYSRSn
9	GPIO9	R/W	0h	Data Register for this pin Reset type: SYSRSn
8	RESERVED	R/W	0h	Reserved
7	GPIO7	R/W	0h	Data Register for this pin Reset type: SYSRSn
6	GPIO6	R/W	0h	Data Register for this pin Reset type: SYSRSn
5	GPIO5	R/W	0h	Data Register for this pin Reset type: SYSRSn
4	GPIO4	R/W	0h	Data Register for this pin Reset type: SYSRSn
3	GPIO3	R/W	0h	Data Register for this pin Reset type: SYSRSn
2	GPIO2	R/W	0h	Data Register for this pin Reset type: SYSRSn
1	GPIO1	R/W	0h	Data Register for this pin Reset type: SYSRSn
0	GPIO0	R/W	0h	Data Register for this pin Reset type: SYSRSn

8.11.3.2 GPASET Register (Offset = 2h) [Reset = 00000000h]

GPASET is shown in Figure 8-43 and described in Table 8-53.

Return to the Summary Table.

GPIO A Data Set Register (GPIO0 to 31)
Writing a 1 will force GPIO output data latch to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 8-43. GPASET Register

31	30	29	28	27	26	25	24
RESERVED	GPIO30	GPIO29	GPIO28	RESERVED	RESERVED	RESERVED	GPIO24
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO23	RESERVED	RESERVED	RESERVED	GPIO19	GPIO18	RESERVED	GPIO16
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 8-53. GPASET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	GPIO30	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
29	GPIO29	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
28	GPIO28	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	GPIO24	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
23	GPIO23	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	GPIO19	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
18	GPIO18	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
17	RESERVED	R-0/W	0h	Reserved
16	GPIO16	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn

Table 8-53. GPASET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	GPIO13	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
12	GPIO12	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
11	GPIO11	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
10	GPIO10	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
9	GPIO9	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
8	RESERVED	R-0/W	0h	Reserved
7	GPIO7	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
6	GPIO6	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
5	GPIO5	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
4	GPIO4	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
3	GPIO3	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
2	GPIO2	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
1	GPIO1	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
0	GPIO0	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn

8.11.3.3 GPACLEAR Register (Offset = 4h) [Reset = 00000000h]

GPACLEAR is shown in Figure 8-44 and described in Table 8-54.

Return to the Summary Table.

GPIO A Data Clear Register (GPIO0 to 31)
Writing a 1 will force GPIO0 output data latch to 0.
Writes of 0 are ignored.
Always reads back a 0.

Figure 8-44. GPACLEAR Register

31	30	29	28	27	26	25	24
RESERVED	GPIO30	GPIO29	GPIO28	RESERVED	RESERVED	RESERVED	GPIO24
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO23	RESERVED	RESERVED	RESERVED	GPIO19	GPIO18	RESERVED	GPIO16
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 8-54. GPACLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	GPIO30	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
29	GPIO29	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
28	GPIO28	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	GPIO24	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
23	GPIO23	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	GPIO19	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
18	GPIO18	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
17	RESERVED	R-0/W	0h	Reserved
16	GPIO16	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn

Table 8-54. GPACLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	GPIO13	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
12	GPIO12	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
11	GPIO11	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
10	GPIO10	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
9	GPIO9	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
8	RESERVED	R-0/W	0h	Reserved
7	GPIO7	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
6	GPIO6	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
5	GPIO5	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
4	GPIO4	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
3	GPIO3	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
2	GPIO2	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
1	GPIO1	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
0	GPIO0	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn

8.11.3.4 GPATOGGLE Register (Offset = 6h) [Reset = 00000000h]

GPATOGGLE is shown in Figure 8-45 and described in Table 8-55.

Return to the Summary Table.

GPIO A Data Toggle Register (GPIO0 to 31)
Writing a 1 will toggle GPIO0 output data latch 1 to 0 or 0 to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 8-45. GPATOGGLE Register

31	30	29	28	27	26	25	24
RESERVED	GPIO30	GPIO29	GPIO28	RESERVED	RESERVED	RESERVED	GPIO24
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
GPIO23	RESERVED	RESERVED	RESERVED	GPIO19	GPIO18	RESERVED	GPIO16
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 8-55. GPATOGGLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	GPIO30	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
29	GPIO29	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
28	GPIO28	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	GPIO24	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
23	GPIO23	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	GPIO19	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
18	GPIO18	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
17	RESERVED	R-0/W	0h	Reserved
16	GPIO16	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn

Table 8-55. GPATOGGLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	GPIO13	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
12	GPIO12	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
11	GPIO11	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
10	GPIO10	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
9	GPIO9	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
8	RESERVED	R-0/W	0h	Reserved
7	GPIO7	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
6	GPIO6	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
5	GPIO5	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
4	GPIO4	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
3	GPIO3	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
2	GPIO2	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
1	GPIO1	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
0	GPIO0	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn

8.11.3.5 GPBDAT Register (Offset = 8h) [Reset = 00000000h]

GPBDAT is shown in Figure 8-46 and described in Table 8-56.

Return to the Summary Table.

GPIO B Data Register (GPIO32 to 63)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register.

Figure 8-46. GPBDAT Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO45	RESERVED	GPIO43	RESERVED	GPIO41	GPIO40
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO39	RESERVED	GPIO37	RESERVED	GPIO35	RESERVED	GPIO33	GPIO32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-56. GPBDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved

Table 8-56. GPBDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	GPIO45	R/W	0h	Data Register for this pin Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	GPIO43	R/W	0h	Data Register for this pin Reset type: SYSRSn
10	RESERVED	R/W	0h	Reserved
9	GPIO41	R/W	0h	Data Register for this pin Reset type: SYSRSn
8	GPIO40	R/W	0h	Data Register for this pin Reset type: SYSRSn
7	GPIO39	R/W	0h	Data Register for this pin Reset type: SYSRSn
6	RESERVED	R/W	0h	Reserved
5	GPIO37	R/W	0h	Data Register for this pin Reset type: SYSRSn
4	RESERVED	R/W	0h	Reserved
3	GPIO35	R/W	0h	Data Register for this pin Reset type: SYSRSn
2	RESERVED	R/W	0h	Reserved
1	GPIO33	R/W	0h	Data Register for this pin Reset type: SYSRSn
0	GPIO32	R/W	0h	Data Register for this pin Reset type: SYSRSn

8.11.3.6 GPBSET Register (Offset = Ah) [Reset = 00000000h]

GPBSET is shown in Figure 8-47 and described in Table 8-57.

Return to the Summary Table.

GPIO B Data Set Register (GPIO32 to 63)
Writing a 1 will force GPIO output data latch to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 8-47. GPBSET Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO45	RESERVED	GPIO43	RESERVED	GPIO41	GPIO40
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO39	RESERVED	GPIO37	RESERVED	GPIO35	RESERVED	GPIO33	GPIO32
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 8-57. GPBSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	RESERVED	R-0/W	0h	Reserved
18	RESERVED	R-0/W	0h	Reserved
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	GPIO45	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
12	RESERVED	R-0/W	0h	Reserved
11	GPIO43	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn

Table 8-57. GPBSET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RESERVED	R-0/W	0h	Reserved
9	GPIO41	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
8	GPIO40	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
7	GPIO39	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
6	RESERVED	R-0/W	0h	Reserved
5	GPIO37	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
4	RESERVED	R-0/W	0h	Reserved
3	GPIO35	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
2	RESERVED	R-0/W	0h	Reserved
1	GPIO33	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
0	GPIO32	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn

8.11.3.7 GPBCLEAR Register (Offset = Ch) [Reset = 00000000h]

GPBCLEAR is shown in Figure 8-48 and described in Table 8-58.

Return to the Summary Table.

GPIO B Data Clear Register (GPIO32 to 63)
Writing a 1 will force GPIO0 output data latch to 0.
Writes of 0 are ignored.
Always reads back a 0.

Figure 8-48. GPBCLEAR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO45	RESERVED	GPIO43	RESERVED	GPIO41	GPIO40
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO39	RESERVED	GPIO37	RESERVED	GPIO35	RESERVED	GPIO33	GPIO32
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 8-58. GPBCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	RESERVED	R-0/W	0h	Reserved
18	RESERVED	R-0/W	0h	Reserved
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	GPIO45	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
12	RESERVED	R-0/W	0h	Reserved
11	GPIO43	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn

Table 8-58. GPBCLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RESERVED	R-0/W	0h	Reserved
9	GPIO41	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
8	GPIO40	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
7	GPIO39	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
6	RESERVED	R-0/W	0h	Reserved
5	GPIO37	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
4	RESERVED	R-0/W	0h	Reserved
3	GPIO35	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
2	RESERVED	R-0/W	0h	Reserved
1	GPIO33	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
0	GPIO32	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn

8.11.3.8 GPBTOGGLE Register (Offset = Eh) [Reset = 00000000h]

GPBTOGGLE is shown in Figure 8-49 and described in Table 8-59.

Return to the Summary Table.

GPIO B Data Toggle Register (GPIO32 to 63)
Writing a 1 will toggle GPIO0 output data latch 1 to 0 or 0 to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 8-49. GPBTOGGLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO45	RESERVED	GPIO43	RESERVED	GPIO41	GPIO40
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
GPIO39	RESERVED	GPIO37	RESERVED	GPIO35	RESERVED	GPIO33	GPIO32
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 8-59. GPBTOGGLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	RESERVED	R-0/W	0h	Reserved
18	RESERVED	R-0/W	0h	Reserved
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	GPIO45	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
12	RESERVED	R-0/W	0h	Reserved
11	GPIO43	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn

Table 8-59. GPBTOGGLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RESERVED	R-0/W	0h	Reserved
9	GPIO41	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
8	GPIO40	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
7	GPIO39	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
6	RESERVED	R-0/W	0h	Reserved
5	GPIO37	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
4	RESERVED	R-0/W	0h	Reserved
3	GPIO35	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
2	RESERVED	R-0/W	0h	Reserved
1	GPIO33	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
0	GPIO32	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn

8.11.3.9 GPHDAT Register (Offset = 38h) [Reset = 00000000h]

GPHDAT is shown in Figure 8-50 and described in Table 8-60.

Return to the Summary Table.

GPIO H Data Register (GPIO224 to 255)
Reading this register indicates the current status of the GPIO pin, irrespective of which mode the pin is in. Writing to this register will set the GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero.
DESIGNER NOTE:
[1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the

Figure 8-50. GPHDAT Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	GPIO245	GPIO244	GPIO243	GPIO242	GPIO241	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	RESERVED	RESERVED	RESERVED	GPIO233	GPIO232
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GPIO231	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-60. GPHDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	GPIO245	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

Table 8-60. GPHDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	GPIO244	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
19	GPIO243	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
18	GPIO242	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
17	GPIO241	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
16	RESERVED	R/W	0h	Reserved
15	GPIO239	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

Table 8-60. GPHDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	GPIO238	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
13	GPIO237	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	GPIO233	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
8	GPIO232	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

Table 8-60. GPHDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	GPIO231	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
6	GPIO230	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
5	RESERVED	R/W	0h	Reserved
4	GPIO228	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
3	GPIO227	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
2	GPIO226	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

Table 8-60. GPHDAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	GPIO225	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn
0	GPIO224	R/W	0h	Reading this register indicates the current status of this GPIO pin, irrespective of which mode the pin is in. Writing to this register will set this GPIO pin high or low if the pin is enabled for GPIO output mode, otherwise the value written is latched but ignored. The state of the output register latch will remain in its current state until the next write operation. A system reset will clear all bits and latched values to zero. DESIGNER NOTE: [1] Reading the GPIODAT register should reflect the state of the PIN (after qualification), not the state of the output latch of the GPIODAT register. Reset type: SYSRSn

8.11.3.10 GPHSET Register (Offset = 3Ah) [Reset = 00000000h]

GPHSET is shown in Figure 8-51 and described in Table 8-61.

Return to the Summary Table.

GPIO H Data Set Register (GPIO224 to 255)
Writing a 1 will force GPIO output data latch to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 8-51. GPHSET Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	GPIO243	GPIO242	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
RESERVED	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	RESERVED	GPIO224
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 8-61. GPHSET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	GPIO243	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
18	GPIO242	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	RESERVED	R-0/W	0h	Reserved
12	RESERVED	R-0/W	0h	Reserved
11	RESERVED	R-0/W	0h	Reserved

Table 8-61. GPHSET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RESERVED	R-0/W	0h	Reserved
9	RESERVED	R-0/W	0h	Reserved
8	RESERVED	R-0/W	0h	Reserved
7	RESERVED	R-0/W	0h	Reserved
6	GPIO230	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
5	RESERVED	R-0/W	0h	Reserved
4	GPIO228	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
3	GPIO227	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
2	GPIO226	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn
1	RESERVED	R-0/W	0h	Reserved
0	GPIO224	R-0/W	0h	Output Set bit for this pin Reset type: SYSRSn

8.11.3.11 GPHCLEAR Register (Offset = 3Ch) [Reset = 00000000h]

GPHCLEAR is shown in Figure 8-52 and described in Table 8-62.

Return to the Summary Table.

GPIO H Data Clear Register (GPIO224 to 255)
Writing a 1 will force GPIO0 output data latch to 0.
Writes of 0 are ignored.
Always reads back a 0.

Figure 8-52. GPHCLEAR Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	GPIO243	GPIO242	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
RESERVED	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	RESERVED	GPIO224
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 8-62. GPHCLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	GPIO243	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
18	GPIO242	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	RESERVED	R-0/W	0h	Reserved
12	RESERVED	R-0/W	0h	Reserved
11	RESERVED	R-0/W	0h	Reserved

Table 8-62. GPHCLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RESERVED	R-0/W	0h	Reserved
9	RESERVED	R-0/W	0h	Reserved
8	RESERVED	R-0/W	0h	Reserved
7	RESERVED	R-0/W	0h	Reserved
6	GPIO230	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
5	RESERVED	R-0/W	0h	Reserved
4	GPIO228	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
3	GPIO227	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
2	GPIO226	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn
1	RESERVED	R-0/W	0h	Reserved
0	GPIO224	R-0/W	0h	Output Clear bit for this pin Reset type: SYSRSn

8.11.3.12 GPHTOGGLE Register (Offset = 3Eh) [Reset = 00000000h]

GPHTOGGLE is shown in Figure 8-53 and described in Table 8-63.

Return to the Summary Table.

GPIO H Data Toggle Register (GPIO224 to 255)
Writing a 1 will toggle GPIO0 output data latch 1 to 0 or 0 to 1.
Writes of 0 are ignored.
Always reads back a 0.

Figure 8-53. GPHTOGGLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	GPIO243	GPIO242	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h
7	6	5	4	3	2	1	0
RESERVED	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	RESERVED	GPIO224
R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h	R-0/W-0h

Table 8-63. GPHTOGGLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W	0h	Reserved
30	RESERVED	R-0/W	0h	Reserved
29	RESERVED	R-0/W	0h	Reserved
28	RESERVED	R-0/W	0h	Reserved
27	RESERVED	R-0/W	0h	Reserved
26	RESERVED	R-0/W	0h	Reserved
25	RESERVED	R-0/W	0h	Reserved
24	RESERVED	R-0/W	0h	Reserved
23	RESERVED	R-0/W	0h	Reserved
22	RESERVED	R-0/W	0h	Reserved
21	RESERVED	R-0/W	0h	Reserved
20	RESERVED	R-0/W	0h	Reserved
19	GPIO243	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
18	GPIO242	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
17	RESERVED	R-0/W	0h	Reserved
16	RESERVED	R-0/W	0h	Reserved
15	RESERVED	R-0/W	0h	Reserved
14	RESERVED	R-0/W	0h	Reserved
13	RESERVED	R-0/W	0h	Reserved
12	RESERVED	R-0/W	0h	Reserved
11	RESERVED	R-0/W	0h	Reserved

Table 8-63. GPHTOGGLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RESERVED	R-0/W	0h	Reserved
9	RESERVED	R-0/W	0h	Reserved
8	RESERVED	R-0/W	0h	Reserved
7	RESERVED	R-0/W	0h	Reserved
6	GPIO230	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
5	RESERVED	R-0/W	0h	Reserved
4	GPIO228	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
3	GPIO227	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
2	GPIO226	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn
1	RESERVED	R-0/W	0h	Reserved
0	GPIO224	R-0/W	0h	Output Toggle Register GPIO pin Reset type: SYSRSn

8.11.4 GPIO_DATA_READ_REGS Registers

Table 8-64 lists the memory-mapped registers for the GPIO_DATA_READ_REGS registers. All register offset addresses not listed in Table 8-64 should be considered as reserved locations and the register contents should not be modified.

Table 8-64. GPIO_DATA_READ_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	GPADAT_R	GPIO A Data Read Register	
2h	GPBDAT_R	GPIO B Data Read Register	
Eh	GPHDAT_R	GPIO H Data Read Register	

Complex bit access types are encoded to fit into small table cells. Table 8-65 shows the codes that are used for access types in this section.

Table 8-65. GPIO_DATA_READ_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

8.11.4.1 GPADAT_R Register (Offset = 0h) [Reset = 00000000h]

GPADAT_R is shown in Figure 8-54 and described in Table 8-66.

Return to the Summary Table.

GPIO A Data Read Register.
Returns the contents of GPADAT register on a read, write to this register has no effect

Figure 8-54. GPADAT_R Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
R-0h																															

Table 8-66. GPADAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Returns the contents of what was written to the GPADAT register on a read, write to this register has no effect Reset type: CPU1.SYSRSn

8.11.4.2 GPBDAT_R Register (Offset = 2h) [Reset = 00000000h]

GPBDAT_R is shown in Figure 8-55 and described in Table 8-67.

Return to the Summary Table.

GPIO B Data Read Register.
Returns the contents of GPBDAT register on a read, write to this register has no effect

Figure 8-55. GPBDAT_R Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
R-0h																															

Table 8-67. GPBDAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Returns the contents of what was written to the GPBDAT register on a read, write to this register has no effect Reset type: CPU1.SYSRSn

8.11.4.3 GPHDAT_R Register (Offset = Eh) [Reset = 00000000h]

GPHDAT_R is shown in Figure 8-56 and described in Table 8-68.

Return to the Summary Table.

GPIO H Data Read Register.
Returns the contents of GPHDAT register on a read, write to this register has no effect

Figure 8-56. GPHDAT_R Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															
R-0h																															

Table 8-68. GPHDAT_R Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Returns the contents of what was written to the GPHDAT register on a read, write to this register has no effect Reset type: CPU1.SYSRSn

Chapter 9

Crossbar (X-BAR)

The crossbars (referred to as X-BAR throughout this chapter) provide flexibility to connect device inputs, outputs, and internal resources in a variety of configurations.

9.1 Input X-BAR

On this device, the Input X-BAR is used to route signals from a GPIO to many different IP blocks such as the ADC, eCAP, MCPWM, and external interrupts. The input of each Input X-BAR instance (INPUTx) can be any GPIO , while the output of each instance connects to various IP blocks in the device. The digital input of AIOs are also available as inputs to the Input X-BAR. This flexibility relieves some of the constraints on peripheral muxing by allowing the user to connect any GPIO to the specified outputs of each Input X-BAR instance. Note that the GPIO selected by the Input X-BAR can be configured as either an input or an output. The Input X-BAR simply connects the signal on the input buffer to the output of the selected Input X-BAR instance. Therefore, you can do things such as route the output of an MCPWM to the eCAP module for a frequency test).

The Input X-BAR is configured by way of the INPUTxSELECT registers. The destinations for each INPUTx are shown in Figure 9-1 and Table 9-1. For additional details on how each Input X-BAR connects to other IP blocks throughout the device, look for references to Input X-BAR in the chapter associated with that IP. Note that the destinations of each INPUTx are fixed and are not user-configurable. For more information on configuring the Input X-BAR, see the INPUT_XBAR_REGS register definitions in the *XBAR Registers* section.

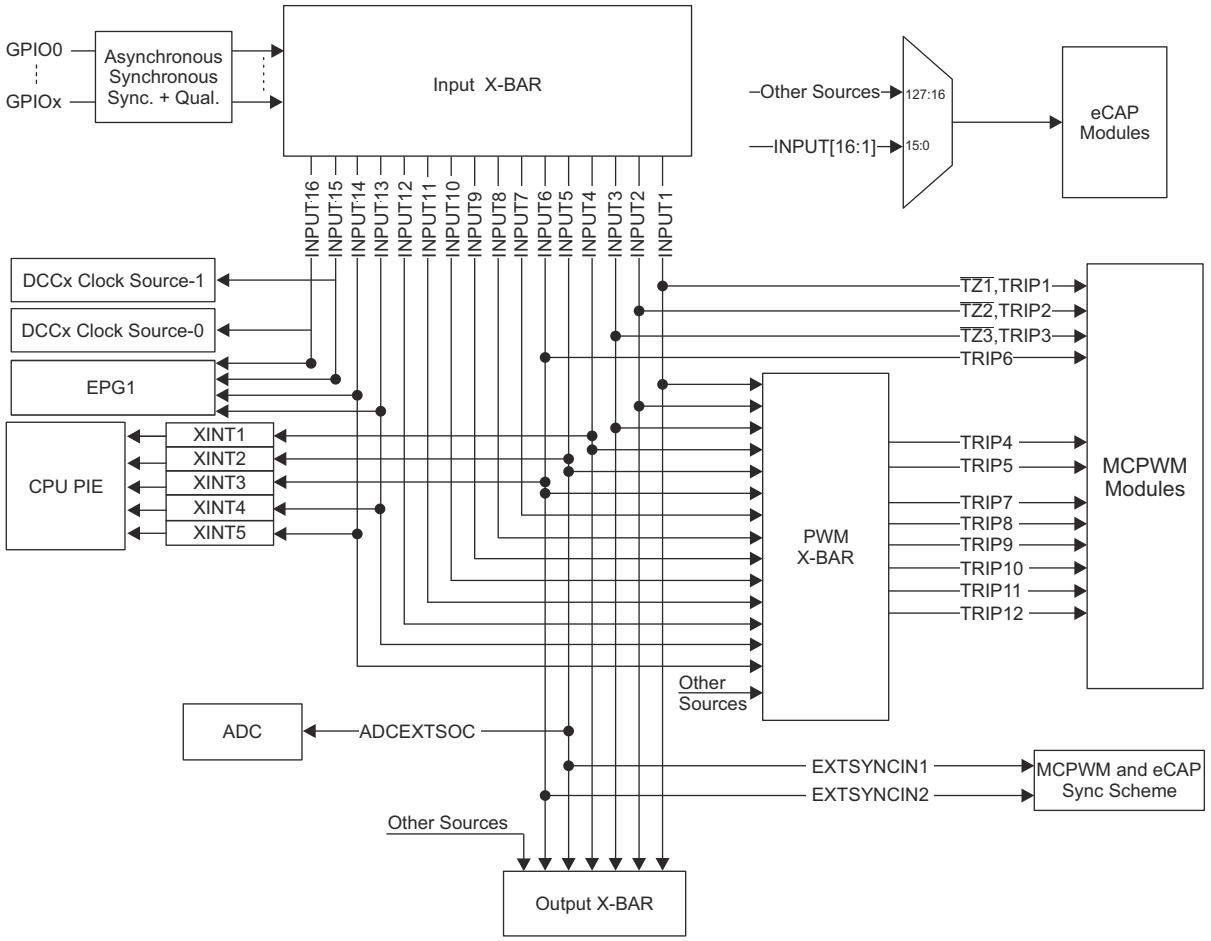


Figure 9-1. Input X-BAR

Note

INPUTXBARx, INPUTXBAR_INPUTx, and INPUTx (when referenced in the context of Input X-BAR) are equivalent in software and documentation.

Table 9-1. Input X-BAR Destinations

INPUT	ECAP	PWM XBAR	OUTPUT XBAR	CPU XINT	ADC START OF CONVERSION	PWM / ECAP SYNC	DCCx	CMPSS
1	Yes	Yes	Yes	-	-	-	-	-
2	Yes	Yes	Yes	-	-	-	-	-
3	Yes	Yes	Yes	-	-	-	-	-
4	Yes	Yes	Yes	XINT1	-	-	-	-
5	Yes	Yes	Yes	XINT2	ADCEXTSOC	EXTSYNCIN1	-	-
6	Yes	Yes	Yes	XINT3	-	EXTSYNCIN2	-	-
7	Yes	Yes	-	-	-	-	-	CMPSS1_EXT_FILTIN_H[1] / CMPSS3_EXT_FILTIN_H[1]
8	Yes	Yes	-	-	-	-	-	CMPSS1_EXT_FILTIN_L[1] / CMPSS3_EXT_FILTIN_L[1]
9	Yes	Yes	-	-	-	-	-	CMPSS2_EXT_FILTIN_H[1] / CMPSS4_EXT_FILTIN_H[1]
10	Yes	Yes	-	-	-	-	-	CMPSS2_EXT_FILTIN_L[1] / CMPSS4_EXT_FILTIN_L[1]
11	Yes	Yes	-	-	-	-	CLK1	-
12	Yes	Yes	-	-	-	-	CLK1	-
13	Yes	Yes	-	XINT4	-	-	-	-
14	Yes	Yes	-	XINT5	-	-	-	-
15	Yes	-	-	-	-	-	CLK1	-
16	Yes	-	-	-	-	-	CLK0	-

9.2 MCPWM and GPIO Output X-BAR

This section describes the MCPWM and GPIO Output X-BAR. .

9.2.1 MCPWM X-BAR

The MCPWM X-BAR brings signals to the MCPWM modules. Specifically, the MCPWM X-BAR outputs are connected to the TZ inputs of the MCPWM modules. Refer to the *Multi-Channel Pulse Width Modulator (MCPWM)* chapter for more information. Figure 9-2 shows the architecture of the MCPWM X-BAR. Note that the architecture of the MCPWM X-BAR is identical to the architecture of the GPIO Output X-BAR (with the exception of the output latch).

9.2.1.1 MCPWM X-BAR Architecture

The MCPWM X-BAR has eight outputs that are routed to each MCPWM module. Figure 9-2 represents the architecture of a single output, but this output is identical to the architecture of all of the other outputs.

First, determine the signals that can be passed to the MCPWM by referencing Table 9-2. Select up to one signal for each TRIPx output. Select the inputs to MCPWM X-BAR using the TRIPxMUX0TO15CFG and TRIPxMUX16TO31CFG registers. To pass any signal through to the MCPWM, enable the signal using the TRIPxMUXENABLE register. All signals that are enabled are logically ORed before being passed on to the respective TRIPx signal on the MCPWM. To optionally invert the signal, use the TRIPOUTINV register.

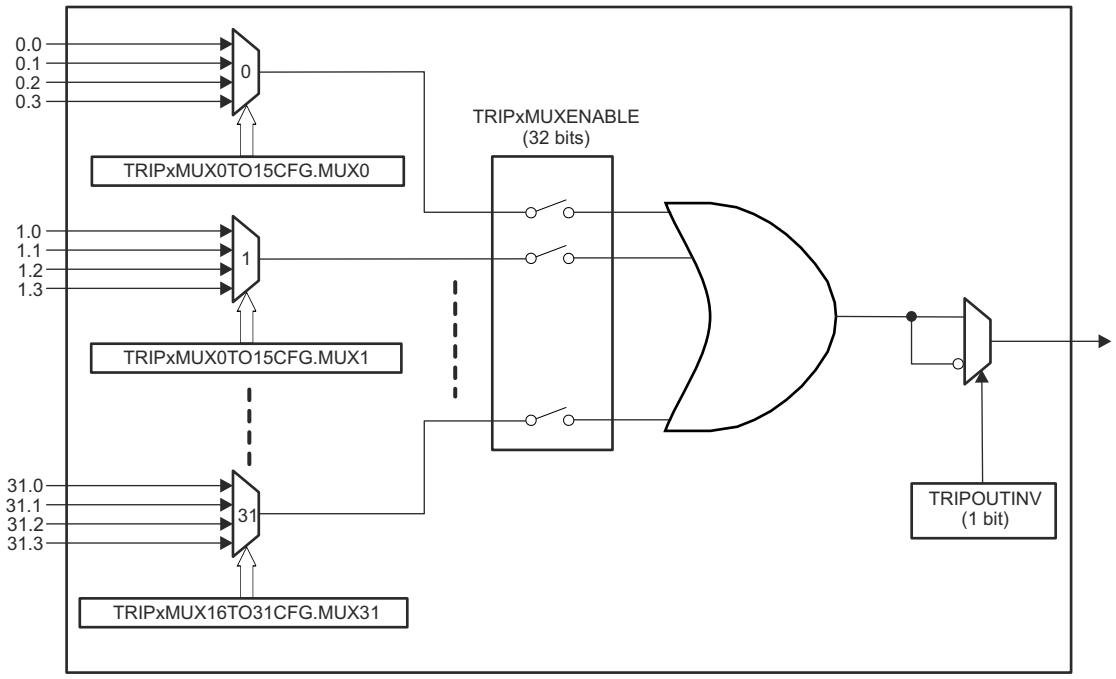


Figure 9-2. MCPWM X-BAR Architecture - Single Output

Note

Do not use "Reserved" signals in your application.

Table 9-2. MCPWM X-BAR Mux Configuration Table

Mux	0	1	2	3
G0	CMPSS1_CTRIPH	CMPSS1_CTRIPH_OR_CTRIPL	ADCAEVT1	ECAP1_OUT
G1	CMPSS1_CTRIPL	INPUTXBAR1	Reserved	ADCCEVT1
G2	CMPSS2_CTRIPH	CMPSS2_CTRIPH_OR_CTRIPL	ADCAEVT2	ECAP2_OUT
G3	CMPSS2_CTRIPL	INPUTXBAR2	Reserved	ADCCEVT2
G4	CMPSS3_CTRIPH	CMPSS3_CTRIPH_OR_CTRIPL	ADCAEVT3	Reserved
G5	CMPSS3_CTRIPL	INPUTXBAR3	Reserved	ADCCEVT3
G6	Reserved	Reserved	Reserved	Reserved
G7	Reserved	INPUTXBAR4	Reserved	Reserved
G8	Reserved	INPUTXBAR5	Reserved	Reserved
G9	Reserved	INPUTXBAR6	Reserved	EMUSTOP
G10	Reserved	INPUTXBAR7	ADCSOCA	Reserved
G11	Reserved	INPUTXBAR8	Reserved	EXTSYNCOUT
G12	Reserved	INPUTXBAR9	ADCSOCB	Reserved
G13	Reserved	INPUTXBAR10	Reserved	EQEP1ERR
G14	Reserved	INPUTXBAR11	CLOCKFAIL	Reserved
G15	Reserved	INPUTXBAR12	Reserved	ECCDBLERR
G16	Reserved	INPUTXBAR13	PIEERR	Reserved
G17	Reserved	INPUTXBAR14	Reserved	ERRORSTS

9.2.2 GPIO Output X-BAR

The GPIO Output X-BAR takes signals from inside the device and brings them out to a GPIO . Figure 9-3 shows the architecture of the GPIO Output X-BAR. The X-BAR contains eight outputs and each contains at least one position on the GPIO mux, denoted as OUTPUTXBARx. The X-BAR allows the selection of a single input or a logical-OR of many inputs.

9.2.2.1 GPIO Output X-BAR Architecture

The GPIO Output X-BAR has eight outputs that are routed to the GPIO module. Figure 9-3 represents the architecture of a single output, but this output is identical to the architecture of all of the other outputs. Note that the architecture of the Output X-BAR (with the exception of the output latch) is similar to the architecture of the MCPWM X-BAR.

First, determine the signals that can be passed to the GPIO by referencing Table 9-3. Select up to one signal per mux for each OUTPUTXBARx output. Select the inputs to each mux using the OUTPUTxMUX0TO15CFG and OUTPUTxMUX16TO31CFG registers. To pass any signal through to the GPIO , enable the mux in the OUTPUTxMUXENABLE register. All muxes that are enabled are logically ORed before being passed on to the respective OUTPUTx signal on the GPIO module. To optionally invert the signal, use the OUTPUTINV register. The final output is only recognized on the GPIO , if the proper OUTPUTx muxing options are selected using the GPIO registers.

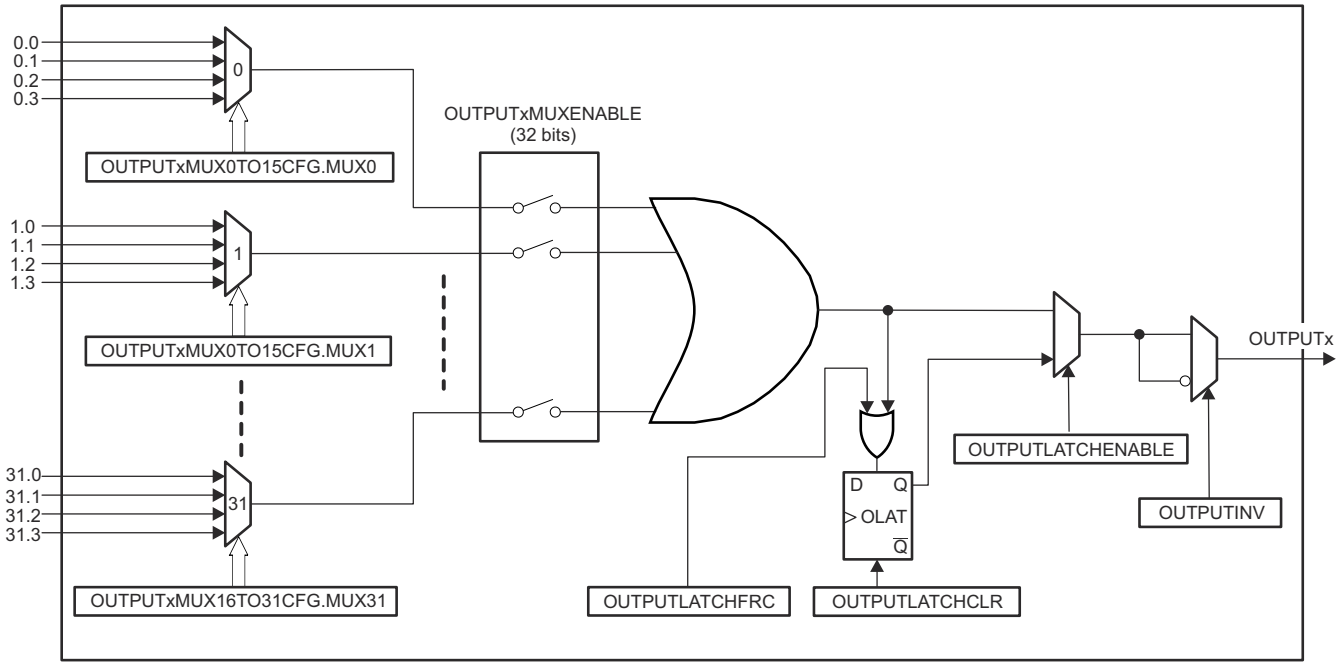


Figure 9-3. GPIO Output X-BAR Architecture

Note

Do not use "Reserved" signals in your application.

Table 9-3. Output X-BAR Mux Configuration Table

Mux	0	1	2	3
G0	CMPSS1_CTRIPOUTH	CMPSS1_CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT1	ECAP1_OUT
G1	CMPSS1_CTRIPOUTL	INPUTXBAR1	Reserved	ADCCEVT1
G2	CMPSS2_CTRIPOUTH	CMPSS2_CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT2	ECAP2_OUT
G3	CMPSS2_CTRIPOUTL	INPUTXBAR2	Reserved	ADCCEVT2
G4	CMPSS3_CTRIPOUTH	CMPSS3_CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT3	Reserved
G5	CMPSS3_CTRIPOUTL	INPUTXBAR3	Reserved	ADCCEVT3
G6	CMPSS4_CTRIPOUTH	CMPSS4_CTRIPOUTH_OR_CTRIPOUTL	Reserved	Reserved
G7	CMPSS4_CTRIPOUTL	INPUTXBAR4	Reserved	Reserved
G8	Reserved	INPUTXBAR5	ADCSOCA	Reserved
G9	Reserved	Reserved	Reserved	EXTSYNCOUT
G10	Reserved	INPUTXBAR6	ADCSOCB	Reserved
G11	Reserved	Reserved	Reserved	ERRORSTS

9.2.3 X-BAR Flags

With the exception of the CMPSS signals, the MCPWM X-BAR and the Output X-BAR have all of the same input signals. Due to the inputs being similar between the MCPWM X-BAR and Output X-BAR, all X-BAR modules leverage a single set of input flags to indicate which input signals have been triggered. This allows software to check the input flags when an event occurs. See Figure 9-4 for more information. There is a bit allocated for each input signal in one of the XBARFLGx registers. The flag remains set until cleared through the appropriate XBARCLRx register.

Note

Not all input sources are routed to all X-BAR modules. Refer to the X-BAR specific configuration tables for exact connections.

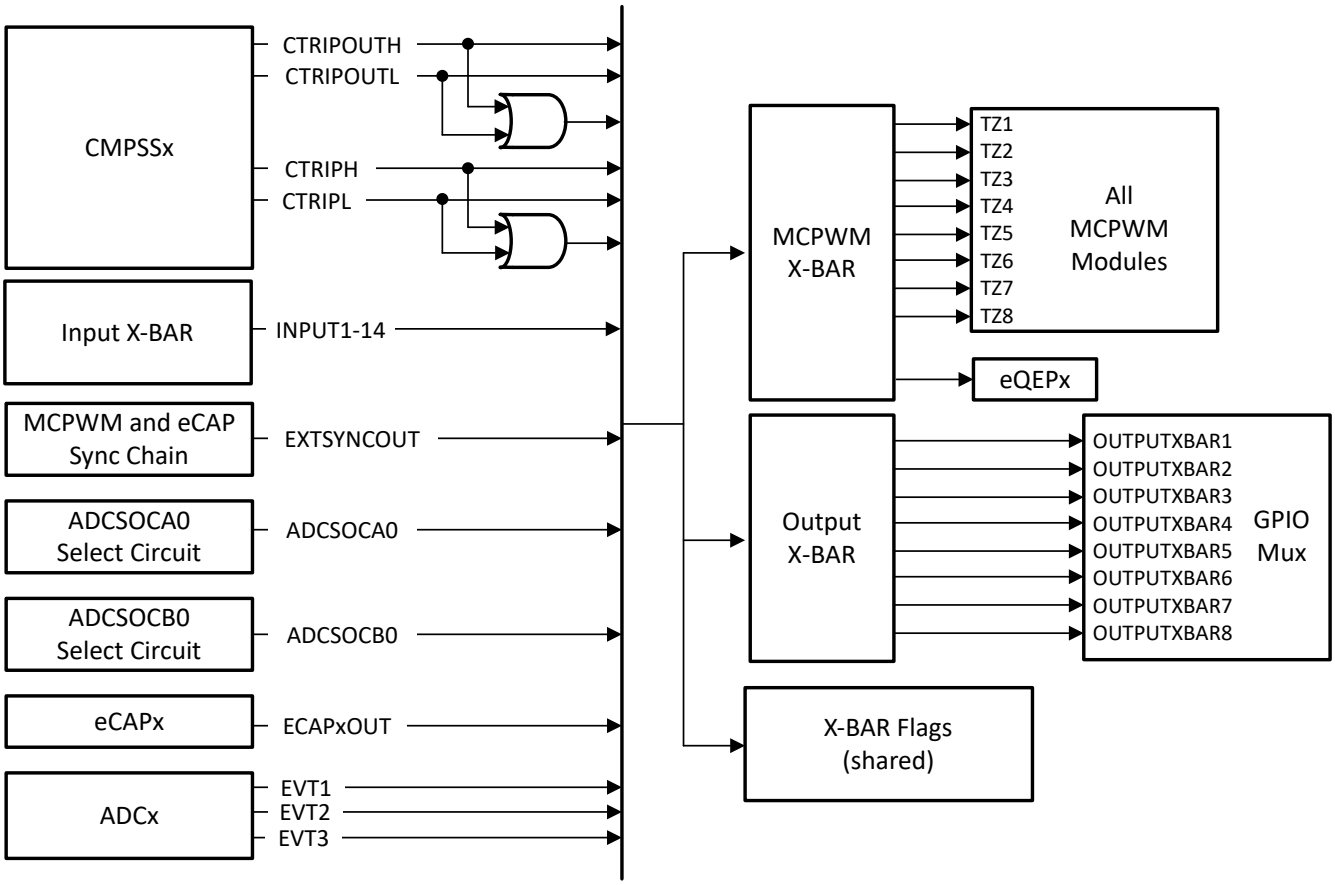


Figure 9-4. X-BAR Input Sources

9.3 XBAR Registers

This section describes the XBAR registers.

9.3.1 XBAR Base Address Table

Table 9-4. XBAR Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
InputXbarRegs	INPUT_XBAR_REGS	INPUTXBAR_BASE	0x0000_7900	-	YES
XbarRegs	XBAR_REGS	XBAR_BASE	0x0000_7920	-	YES
PwmXbarRegs	PWM_XBAR_REGS	PWMXBAR_BASE	0x0000_7A00	-	YES
OutputXbarRegs	OUTPUT_XBAR_REGS	OUTPUTXBAR_BASE	0x0000_7A80	-	YES

9.3.2 INPUT_XBAR_REGS Registers

Table 9-5 lists the memory-mapped registers for the INPUT_XBAR_REGS registers. All register offset addresses not listed in Table 9-5 should be considered as reserved locations and the register contents should not be modified.

Table 9-5. INPUT_XBAR_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	INPUT1SELECT	INPUT1 Input Select Register (GPIO0 to x)	EALLOW
1h	INPUT2SELECT	INPUT2 Input Select Register (GPIO0 to x)	EALLOW
2h	INPUT3SELECT	INPUT3 Input Select Register (GPIO0 to x)	EALLOW
3h	INPUT4SELECT	INPUT4 Input Select Register (GPIO0 to x)	EALLOW
4h	INPUT5SELECT	INPUT5 Input Select Register (GPIO0 to x)	EALLOW
5h	INPUT6SELECT	INPUT6 Input Select Register (GPIO0 to x)	EALLOW
6h	INPUT7SELECT	INPUT7 Input Select Register (GPIO0 to x)	EALLOW
7h	INPUT8SELECT	INPUT8 Input Select Register (GPIO0 to x)	EALLOW
8h	INPUT9SELECT	INPUT9 Input Select Register (GPIO0 to x)	EALLOW
9h	INPUT10SELECT	INPUT10 Input Select Register (GPIO0 to x)	EALLOW
Ah	INPUT11SELECT	INPUT11 Input Select Register (GPIO0 to x)	EALLOW
Bh	INPUT12SELECT	INPUT12 Input Select Register (GPIO0 to x)	EALLOW
Ch	INPUT13SELECT	INPUT13 Input Select Register (GPIO0 to x)	EALLOW
Dh	INPUT14SELECT	INPUT14 Input Select Register (GPIO0 to x)	EALLOW
Eh	INPUT15SELECT	INPUT15 Input Select Register (GPIO0 to x)	EALLOW
Fh	INPUT16SELECT	INPUT16 Input Select Register (GPIO0 to x)	EALLOW
1Eh	INPUTSELECTLOCK	Input Select Lock Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 9-6 shows the codes that are used for access types in this section.

Table 9-6. INPUT_XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

9.3.2.1 INPUT1SELECT Register (Offset = 0h) [Reset = 01FEh]

INPUT1SELECT is shown in Figure 9-5 and described in Table 9-7.

Return to the Summary Table.

INPUT1 Input Select Register (GPIO0 to x)

Figure 9-5. INPUT1SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-7. INPUT1SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT1 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.2 INPUT2SELECT Register (Offset = 1h) [Reset = 01FEh]

INPUT2SELECT is shown in Figure 9-6 and described in Table 9-8.

Return to the Summary Table.

INPUT2 Input Select Register (GPIO0 to x)

Figure 9-6. INPUT2SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-8. INPUT2SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT2 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.3 INPUT3SELECT Register (Offset = 2h) [Reset = 01FEh]

INPUT3SELECT is shown in Figure 9-7 and described in Table 9-9.

Return to the Summary Table.

INPUT3 Input Select Register (GPIO0 to x)

Figure 9-7. INPUT3SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-9. INPUT3SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT3 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.4 INPUT4SELECT Register (Offset = 3h) [Reset = 01FEh]

INPUT4SELECT is shown in Figure 9-8 and described in Table 9-10.

Return to the Summary Table.

INPUT4 Input Select Register (GPIO0 to x)

Figure 9-8. INPUT4SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-10. INPUT4SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT4 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.5 INPUT5SELECT Register (Offset = 4h) [Reset = 01FEh]

INPUT5SELECT is shown in Figure 9-9 and described in Table 9-11.

Return to the Summary Table.

INPUT5 Input Select Register (GPIO0 to x)

Figure 9-9. INPUT5SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-11. INPUT5SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT5 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.6 INPUT6SELECT Register (Offset = 5h) [Reset = 01FEh]

INPUT6SELECT is shown in Figure 9-10 and described in Table 9-12.

Return to the Summary Table.

INPUT6 Input Select Register (GPIO0 to x)

Figure 9-10. INPUT6SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-12. INPUT6SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT6 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.7 INPUT7SELECT Register (Offset = 6h) [Reset = 01FEh]

INPUT7SELECT is shown in Figure 9-11 and described in Table 9-13.

Return to the Summary Table.

INPUT7 Input Select Register (GPIO0 to x)

Figure 9-11. INPUT7SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-13. INPUT7SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT7 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.8 INPUT8SELECT Register (Offset = 7h) [Reset = 01FEh]

INPUT8SELECT is shown in Figure 9-12 and described in Table 9-14.

Return to the Summary Table.

INPUT8 Input Select Register (GPIO0 to x)

Figure 9-12. INPUT8SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-14. INPUT8SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT8 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.9 INPUT9SELECT Register (Offset = 8h) [Reset = 01FEh]

INPUT9SELECT is shown in Figure 9-13 and described in Table 9-15.

Return to the Summary Table.

INPUT9 Input Select Register (GPIO0 to x)

Figure 9-13. INPUT9SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-15. INPUT9SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT9 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.10 INPUT10SELECT Register (Offset = 9h) [Reset = 01FEh]

INPUT10SELECT is shown in Figure 9-14 and described in Table 9-16.

Return to the Summary Table.

INPUT10 Input Select Register (GPIO0 to x)

Figure 9-14. INPUT10SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-16. INPUT10SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT10 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.11 INPUT11SELECT Register (Offset = Ah) [Reset = 01FEh]

INPUT11SELECT is shown in Figure 9-15 and described in Table 9-17.

Return to the Summary Table.

INPUT11 Input Select Register (GPIO0 to x)

Figure 9-15. INPUT11SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-17. INPUT11SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT11 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.12 INPUT12SELECT Register (Offset = Bh) [Reset = 01FEh]

INPUT12SELECT is shown in Figure 9-16 and described in Table 9-18.

Return to the Summary Table.

INPUT12 Input Select Register (GPIO0 to x)

Figure 9-16. INPUT12SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-18. INPUT12SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT12 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.13 INPUT13SELECT Register (Offset = Ch) [Reset = 01FEh]

INPUT13SELECT is shown in Figure 9-17 and described in Table 9-19.

Return to the Summary Table.

INPUT13 Input Select Register (GPIO0 to x)

Figure 9-17. INPUT13SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-19. INPUT13SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT13 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.14 INPUT14SELECT Register (Offset = Dh) [Reset = 01FEh]

INPUT14SELECT is shown in Figure 9-18 and described in Table 9-20.

Return to the Summary Table.

INPUT14 Input Select Register (GPIO0 to x)

Figure 9-18. INPUT14SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-20. INPUT14SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT14 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.15 INPUT15SELECT Register (Offset = Eh) [Reset = 01FEh]

INPUT15SELECT is shown in Figure 9-19 and described in Table 9-21.

Return to the Summary Table.

INPUT15 Input Select Register (GPIO0 to x)

Figure 9-19. INPUT15SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-21. INPUT15SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT15 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.16 INPUT16SELECT Register (Offset = Fh) [Reset = 01FEh]

INPUT16SELECT is shown in Figure 9-20 and described in Table 9-22.

Return to the Summary Table.

INPUT16 Input Select Register (GPIO0 to x)

Figure 9-20. INPUT16SELECT Register

15	14	13	12	11	10	9	8
RESERVED							SELECT
R-0-0h							R/W-1FEh
7	6	5	4	3	2	1	0
SELECT							
R/W-1FEh							

Table 9-22. INPUT16SELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8-0	SELECT	R/W	1FEh	Select GPIO for INPUT16 signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0x1FD: '1' will be driven to the destination 0x1FE: '1' will be driven to the destination 0x1FF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0x1FF) will cause the destination to be driven '1'. Reset type: CPU1.SYSRSn

9.3.2.17 INPUTSELECTLOCK Register (Offset = 1Eh) [Reset = 00000000h]

INPUTSELECTLOCK is shown in Figure 9-21 and described in Table 9-23.

Return to the Summary Table.

Input Select Lock Register.
Any bit in this register, once set can only be cleared through SYSRSn. Write of 0 to any bit of this register has no effect. Reads to the registers which have LOCK protection are always allowed.

Figure 9-21. INPUTSELECTLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
INPUT16SELECT	INPUT15SELECT	INPUT14SELECT	INPUT13SELECT	INPUT12SELECT	INPUT11SELECT	INPUT10SELECT	INPUT9SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
INPUT8SELECT	INPUT7SELECT	INPUT6SELECT	INPUT5SELECT	INPUT4SELECT	INPUT3SELECT	INPUT2SELECT	INPUT1SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 9-23. INPUTSELECTLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	INPUT16SELECT	R/WSonce	0h	Lock bit for INPUT16SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
14	INPUT15SELECT	R/WSonce	0h	Lock bit for INPUT15SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
13	INPUT14SELECT	R/WSonce	0h	Lock bit for INPUT14SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
12	INPUT13SELECT	R/WSonce	0h	Lock bit for INPUT13SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
11	INPUT12SELECT	R/WSonce	0h	Lock bit for INPUT12SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
10	INPUT11SELECT	R/WSonce	0h	Lock bit for INPUT11SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

Table 9-23. INPUTSELECTLOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	INPUT10SELECT	R/WSonce	0h	Lock bit for INPUT10SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
8	INPUT9SELECT	R/WSonce	0h	Lock bit for INPUT9SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
7	INPUT8SELECT	R/WSonce	0h	Lock bit for INPUT8SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
6	INPUT7SELECT	R/WSonce	0h	Lock bit for INPUT7SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
5	INPUT6SELECT	R/WSonce	0h	Lock bit for INPUT6SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
4	INPUT5SELECT	R/WSonce	0h	Lock bit for INPUT5SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
3	INPUT4SELECT	R/WSonce	0h	Lock bit for INPUT4SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
2	INPUT3SELECT	R/WSonce	0h	Lock bit for INPUT3SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
1	INPUT2SELECT	R/WSonce	0h	Lock bit for INPUT2SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn
0	INPUT1SELECT	R/WSonce	0h	Lock bit for INPUT1SELECT Register 0: Register is not locked 1: Register is locked Reset type: CPU1.SYSRSn

9.3.3 XBAR_REGS Registers

Table 9-24 lists the memory-mapped registers for the XBAR_REGS registers. All register offset addresses not listed in Table 9-24 should be considered as reserved locations and the register contents should not be modified.

Table 9-24. XBAR_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	XBARFLG1	X-Bar Input Flag Register 1	
2h	XBARFLG2	X-Bar Input Flag Register 2	
4h	XBARFLG3	X-Bar Input Flag Register 3	
6h	XBARFLG4	X-Bar Input Flag Register 4	
8h	XBARFLG5	X-Bar Input Flag Register 5	
Ah	XBARCLR1	X-Bar Input Flag Clear Register 1	
Ch	XBARCLR2	X-Bar Input Flag Clear Register 2	
Eh	XBARCLR3	X-Bar Input Flag Clear Register 3	
10h	XBARCLR4	X-Bar Input Flag Clear Register 4	
12h	XBARCLR5	X-Bar Input Flag Clear Register 5	

Complex bit access types are encoded to fit into small table cells. Table 9-25 shows the codes that are used for access types in this section.

Table 9-25. XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W1C	W 1C	Write 1 to clear
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

9.3.3.1 XBARFLG1 Register (Offset = 0h) [Reset = 00000000h]

XBARFLG1 is shown in Figure 9-22 and described in Table 9-26.

Return to the Summary Table.

X-Bar Input Flag Register 1

Figure 9-22. XBARFLG1 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	CMPSS3_CTRIPOUTH	CMPSS3_CTRIPOUTL	CMPSS2_CTRIPOUTH	CMPSS2_CTRIPOUTL	CMPSS1_CTRIPOUTH	CMPSS1_CTRIPOUTL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CMPSS3_CTRIPH	CMPSS3_CTRIPL	CMPSS2_CTRIPH	CMPSS2_CTRIPL	CMPSS1_CTRIPH	CMPSS1_CTRIPL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-26. XBARFLG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	CMPSS3_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
20	CMPSS3_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 9-26. XBARFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CMPSS2_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	CMPSS2_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	CMPSS1_CTRIPOUTH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	CMPSS1_CTRIPOUTL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CMPSS3_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	CMPSS3_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 9-26. XBARFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CMPSS2_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	CMPSS2_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	CMPSS1_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	CMPSS1_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

9.3.3.2 XBARFLG2 Register (Offset = 2h) [Reset = 00000000h]

XBARFLG2 is shown in Figure 9-23 and described in Table 9-27.

Return to the Summary Table.

X-Bar Input Flag Register 2

Figure 9-23. XBARFLG2 Register

31	30	29	28	27	26	25	24
ADCCEVT1	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCAEVT3	ADCAEVT2
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
ADCAEVT1	EXTSYNCOUT	RESERVED	RESERVED	RESERVED	RESERVED	ECAP2_OUT	ECAP1_OUT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8	INPUT7
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
ADCSOCB	ADCSOCA	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-27. XBARFLG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ADCCEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	ADCAEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	ADCAEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 9-27. XBARFLG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	ADCAEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	EXTSYNCOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	ECAP2_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	ECAP1_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	INPUT14	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	INPUT13	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	INPUT12	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 9-27. XBARFLG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	INPUT11	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	INPUT10	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	INPUT9	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	INPUT8	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	INPUT7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	ADCSOCB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	ADCSOCA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	INPUT6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 9-27. XBARFLG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	INPUT5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	INPUT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	INPUT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	INPUT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	INPUT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

9.3.3.3 XBARFLG3 Register (Offset = 4h) [Reset = 00000000h]

XBARFLG3 is shown in Figure 9-24 and described in Table 9-28.

Return to the Summary Table.

X-Bar Input Flag Register 3

Figure 9-24. XBARFLG3 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCEVT3	ADCCEVT2
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-28. XBARFLG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	RESERVED	R	0h	Reserved
16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved

Table 9-28. XBARFLG3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	ADCCEVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	ADCCEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

9.3.3.4 XBARFLG4 Register (Offset = 6h) [Reset = 00000000h]

XBARFLG4 is shown in Figure 9-25 and described in Table 9-29.

Return to the Summary Table.

X-Bar Input Flag Register 4

Figure 9-25. XBARFLG4 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	ERRORSTS_ERROR	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-29. XBARFLG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	RESERVED	R	0h	Reserved
29	RESERVED	R	0h	Reserved
28	ERRORSTS_ERROR	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: ERRORSTS_ERROR input was triggered 0: ERRORSTS_ERROR Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	RESERVED	R	0h	Reserved
24	RESERVED	R	0h	Reserved
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	RESERVED	R	0h	Reserved
20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	RESERVED	R	0h	Reserved
16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved

Table 9-29. XBARFLG4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

9.3.3.5 XBARFLG5 Register (Offset = 8h) [Reset = 00000000h]

XBARFLG5 is shown in Figure 9-26 and described in Table 9-30.

Return to the Summary Table.

X-Bar Input Flag Register 5

Figure 9-26. XBARFLG5 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		PIEERR	ECCDBLERR	CLOCKFAIL	RESERVED	EQEP1ERR	EMUSTOP
R-0-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-30. XBARFLG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	PIEERR	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	ECCDBLERR	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	CLOCKFAIL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	RESERVED	R	0h	Reserved
1	EQEP1ERR	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 9-30. XBARFLG5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	EMUSTOP	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

9.3.3.6 XBARCLR1 Register (Offset = Ah) [Reset = 00000000h]

XBARCLR1 is shown in Figure 9-27 and described in Table 9-31.

Return to the Summary Table.

X-Bar Input Flag Clear Register 1

Figure 9-27. XBARCLR1 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	CMPSS3_CTRIPOUTH	CMPSS3_CTRIPOUTL	CMPSS2_CTRIPOUTH	CMPSS2_CTRIPOUTL	CMPSS1_CTRIPOUTH	CMPSS1_CTRIPOUTL
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CMPSS3_CTRIPH	CMPSS3_CTRIPL	CMPSS2_CTRIPH	CMPSS2_CTRIPL	CMPSS1_CTRIPH	CMPSS1_CTRIPL
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 9-31. XBARCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W1C	0h	Reserved
30	RESERVED	R-0/W1C	0h	Reserved
29	RESERVED	R-0/W1C	0h	Reserved
28	RESERVED	R-0/W1C	0h	Reserved
27	RESERVED	R-0/W1C	0h	Reserved
26	RESERVED	R-0/W1C	0h	Reserved
25	RESERVED	R-0/W1C	0h	Reserved
24	RESERVED	R-0/W1C	0h	Reserved
23	RESERVED	R-0/W1C	0h	Reserved
22	RESERVED	R-0/W1C	0h	Reserved
21	CMPSS3_CTRIPOUTH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	CMPSS3_CTRIPOUTL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	CMPSS2_CTRIPOUTH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	CMPSS2_CTRIPOUTL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 9-31. XBARCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	CMPSS1_CTRIPOUTH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	CMPSS1_CTRIPOUTL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	RESERVED	R-0/W1C	0h	Reserved
14	RESERVED	R-0/W1C	0h	Reserved
13	RESERVED	R-0/W1C	0h	Reserved
12	RESERVED	R-0/W1C	0h	Reserved
11	RESERVED	R-0/W1C	0h	Reserved
10	RESERVED	R-0/W1C	0h	Reserved
9	RESERVED	R-0/W1C	0h	Reserved
8	RESERVED	R-0/W1C	0h	Reserved
7	RESERVED	R-0/W1C	0h	Reserved
6	RESERVED	R-0/W1C	0h	Reserved
5	CMPSS3_CTRIPH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	CMPSS3_CTRIPL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	CMPSS2_CTRIPH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	CMPSS2_CTRIPL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	CMPSS1_CTRIPH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	CMPSS1_CTRIPL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

9.3.3.7 XBARCLR2 Register (Offset = Ch) [Reset = 00000000h]

XBARCLR2 is shown in Figure 9-28 and described in Table 9-32.

Return to the Summary Table.

X-Bar Input Flag Clear Register 2

Figure 9-28. XBARCLR2 Register

31	30	29	28	27	26	25	24
ADCCEVT1	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCAEVT3	ADCAEVT2
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
23	22	21	20	19	18	17	16
ADCAEVT1	EXTSYNCOUT	RESERVED	RESERVED	RESERVED	RESERVED	ECAP2_OUT	ECAP1_OUT
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8	INPUT7
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
ADCSOCB	ADCSOCA	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 9-32. XBARCLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ADCCEVT1	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	RESERVED	R-0/W1C	0h	Reserved
29	RESERVED	R-0/W1C	0h	Reserved
28	RESERVED	R-0/W1C	0h	Reserved
27	RESERVED	R-0/W1C	0h	Reserved
26	RESERVED	R-0/W1C	0h	Reserved
25	ADCAEVT3	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
24	ADCAEVT2	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
23	ADCAEVT1	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
22	EXTSYNCOUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	RESERVED	R-0/W1C	0h	Reserved
20	RESERVED	R-0/W1C	0h	Reserved
19	RESERVED	R-0/W1C	0h	Reserved
18	RESERVED	R-0/W1C	0h	Reserved

Table 9-32. XBARCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	ECAP2_OUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	ECAP1_OUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	INPUT14	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
14	INPUT13	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	INPUT12	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	INPUT11	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	INPUT10	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	INPUT9	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	INPUT8	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	INPUT7	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	ADCSOCB	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	ADCSOCA	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	INPUT6	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	INPUT5	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 9-32. XBARCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INPUT4	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	INPUT3	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	INPUT2	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	INPUT1	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

9.3.3.8 XBARCLR3 Register (Offset = Eh) [Reset = 00000000h]

XBARCLR3 is shown in Figure 9-29 and described in Table 9-33.

Return to the Summary Table.

X-Bar Input Flag Clear Register 3

Figure 9-29. XBARCLR3 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCEVT3	ADCCEVT2
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 9-33. XBARCLR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W1C	0h	Reserved
30	RESERVED	R-0/W1C	0h	Reserved
29	RESERVED	R-0/W1C	0h	Reserved
28	RESERVED	R-0/W1C	0h	Reserved
27	RESERVED	R-0/W1C	0h	Reserved
26	RESERVED	R-0/W1C	0h	Reserved
25	RESERVED	R-0/W1C	0h	Reserved
24	RESERVED	R-0/W1C	0h	Reserved
23	RESERVED	R-0/W1C	0h	Reserved
22	RESERVED	R-0/W1C	0h	Reserved
21	RESERVED	R-0/W1C	0h	Reserved
20	RESERVED	R-0/W1C	0h	Reserved
19	RESERVED	R-0/W1C	0h	Reserved
18	RESERVED	R-0/W1C	0h	Reserved
17	RESERVED	R-0/W1C	0h	Reserved
16	RESERVED	R-0/W1C	0h	Reserved
15	RESERVED	R-0/W1C	0h	Reserved
14	RESERVED	R-0/W1C	0h	Reserved
13	RESERVED	R-0/W1C	0h	Reserved
12	RESERVED	R-0/W1C	0h	Reserved
11	RESERVED	R-0/W1C	0h	Reserved
10	RESERVED	R-0/W1C	0h	Reserved
9	RESERVED	R-0/W1C	0h	Reserved
8	RESERVED	R-0/W1C	0h	Reserved
7	RESERVED	R-0/W1C	0h	Reserved

Table 9-33. XBARCLR3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RESERVED	R-0/W1C	0h	Reserved
5	RESERVED	R-0/W1C	0h	Reserved
4	RESERVED	R-0/W1C	0h	Reserved
3	RESERVED	R-0/W1C	0h	Reserved
2	RESERVED	R-0/W1C	0h	Reserved
1	ADCCEVT3	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	ADCCEVT2	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG3 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

9.3.3.9 XBARCLR4 Register (Offset = 10h) [Reset = 00000000h]

XBARCLR4 is shown in Figure 9-30 and described in Table 9-34.

Return to the Summary Table.

X-Bar Input Flag Clear Register 4

Figure 9-30. XBARCLR4 Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	ERRORSTS_ERROR	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 9-34. XBARCLR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R-0/W1C	0h	Reserved
30	RESERVED	R-0/W1C	0h	Reserved
29	RESERVED	R-0/W1C	0h	Reserved
28	ERRORSTS_ERROR	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG4 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	RESERVED	R-0/W1C	0h	Reserved
26	RESERVED	R-0/W1C	0h	Reserved
25	RESERVED	R-0/W1C	0h	Reserved
24	RESERVED	R-0/W1C	0h	Reserved
23	RESERVED	R-0/W1C	0h	Reserved
22	RESERVED	R-0/W1C	0h	Reserved
21	RESERVED	R-0/W1C	0h	Reserved
20	RESERVED	R-0/W1C	0h	Reserved
19	RESERVED	R-0/W1C	0h	Reserved
18	RESERVED	R-0/W1C	0h	Reserved
17	RESERVED	R-0/W1C	0h	Reserved
16	RESERVED	R-0/W1C	0h	Reserved
15	RESERVED	R-0/W1C	0h	Reserved
14	RESERVED	R-0/W1C	0h	Reserved
13	RESERVED	R-0/W1C	0h	Reserved
12	RESERVED	R-0/W1C	0h	Reserved
11	RESERVED	R-0/W1C	0h	Reserved
10	RESERVED	R-0/W1C	0h	Reserved

Table 9-34. XBARCLR4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	RESERVED	R-0/W1C	0h	Reserved
8	RESERVED	R-0/W1C	0h	Reserved
7	RESERVED	R-0/W1C	0h	Reserved
6	RESERVED	R-0/W1C	0h	Reserved
5	RESERVED	R-0/W1C	0h	Reserved
4	RESERVED	R-0/W1C	0h	Reserved
3	RESERVED	R-0/W1C	0h	Reserved
2	RESERVED	R-0/W1C	0h	Reserved
1	RESERVED	R-0/W1C	0h	Reserved
0	RESERVED	R-0/W1C	0h	Reserved

9.3.3.10 XBARCLR5 Register (Offset = 12h) [Reset = 00000000h]

XBARCLR5 is shown in Figure 9-31 and described in Table 9-35.

Return to the Summary Table.

X-Bar Input Flag Clear Register 5

Figure 9-31. XBARCLR5 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		PIEERR	ECCDBLERR	CLOCKFAIL	RESERVED	EQEP1ERR	EMUSTOP
R-0-0h		R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 9-35. XBARCLR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	PIEERR	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	ECCDBLERR	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
3	CLOCKFAIL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	RESERVED	R-0/W1C	0h	Reserved
1	EQEP1ERR	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	EMUSTOP	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG5 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

9.3.4 PWM_XBAR_REGS Registers

Table 9-36 lists the memory-mapped registers for the PWM_XBAR_REGS registers. All register offset addresses not listed in Table 9-36 should be considered as reserved locations and the register contents should not be modified.

Table 9-36. PWM_XBAR_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	TRIP1MUX0TO15CFG	PWM XBAR Mux Configuration for Output1	EALLOW
2h	TRIP1MUX16TO31CFG	PWM XBAR Mux Configuration for Output1	EALLOW
4h	TRIP2MUX0TO15CFG	PWM XBAR Mux Configuration for Output2	EALLOW
6h	TRIP2MUX16TO31CFG	PWM XBAR Mux Configuration for Output2	EALLOW
8h	TRIP3MUX0TO15CFG	PWM XBAR Mux Configuration for Output3	EALLOW
Ah	TRIP3MUX16TO31CFG	PWM XBAR Mux Configuration for Output3	EALLOW
Ch	TRIP4MUX0TO15CFG	PWM XBAR Mux Configuration for Output4	EALLOW
Eh	TRIP4MUX16TO31CFG	PWM XBAR Mux Configuration for Output4	EALLOW
10h	TRIP5MUX0TO15CFG	PWM XBAR Mux Configuration for Output5	EALLOW
12h	TRIP5MUX16TO31CFG	PWM XBAR Mux Configuration for Output5	EALLOW
14h	TRIP6MUX0TO15CFG	PWM XBAR Mux Configuration for Output6	EALLOW
16h	TRIP6MUX16TO31CFG	PWM XBAR Mux Configuration for Output6	EALLOW
18h	TRIP7MUX0TO15CFG	PWM XBAR Mux Configuration for Output7	EALLOW
1Ah	TRIP7MUX16TO31CFG	PWM XBAR Mux Configuration for Output7	EALLOW
1Ch	TRIP8MUX0TO15CFG	PWM XBAR Mux Configuration for Output8	EALLOW
1Eh	TRIP8MUX16TO31CFG	PWM XBAR Mux Configuration for Output8	EALLOW
20h	TRIP1MUXENABLE	PWM XBAR Mux Enable for Output1	EALLOW
22h	TRIP2MUXENABLE	PWM XBAR Mux Enable for Output2	EALLOW
24h	TRIP3MUXENABLE	PWM XBAR Mux Enable for Output3	EALLOW
26h	TRIP4MUXENABLE	PWM XBAR Mux Enable for Output4	EALLOW
28h	TRIP5MUXENABLE	PWM XBAR Mux Enable for Output5	EALLOW
2Ah	TRIP6MUXENABLE	PWM XBAR Mux Enable for Output6	EALLOW
2Ch	TRIP7MUXENABLE	PWM XBAR Mux Enable for Output7	EALLOW
2Eh	TRIP8MUXENABLE	PWM XBAR Mux Enable for Output8	EALLOW
38h	TRIPOUTINV	PWM XBAR Output Inversion Register	EALLOW
3Eh	TRIPLOCK	PWM XBAR Configuration Lock register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 9-37 shows the codes that are used for access types in this section.

Table 9-37. PWM_XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value

Table 9-37. PWM_XBAR_REGS Access Type Codes (continued)

Access Type	Code	Description
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

9.3.4.1 TRIP1MUX0TO15CFG Register (Offset = 0h) [Reset = 00000000h]

TRIP1MUX0TO15CFG is shown in Figure 9-32 and described in Table 9-38.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output1

Figure 9-32. TRIP1MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15		MUX14		MUX13		MUX12		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-38. TRIP1MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	MUX15	R/W	0h	Select Bits for PWM-XBAR OUT1MUX15: 00 : Select .0 input for MUX15 01 : Select .1 input for MUX15 10 : Select .2 input for MUX15 11 : Select .3 input for MUX15 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
29-28	MUX14	R/W	0h	Select Bits for PWM-XBAR OUT1MUX14: 00 : Select .0 input for MUX14 01 : Select .1 input for MUX14 10 : Select .2 input for MUX14 11 : Select .3 input for MUX14 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
27-26	MUX13	R/W	0h	Select Bits for PWM-XBAR OUT1MUX13: 00 : Select .0 input for MUX13 01 : Select .1 input for MUX13 10 : Select .2 input for MUX13 11 : Select .3 input for MUX13 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
25-24	MUX12	R/W	0h	Select Bits for PWM-XBAR OUT1MUX12: 00 : Select .0 input for MUX12 01 : Select .1 input for MUX12 10 : Select .2 input for MUX12 11 : Select .3 input for MUX12 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
23-22	MUX11	R/W	0h	Select Bits for PWM-XBAR OUT1MUX11: 00 : Select .0 input for MUX11 01 : Select .1 input for MUX11 10 : Select .2 input for MUX11 11 : Select .3 input for MUX11 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for PWM-XBAR OUT1MUX10: 00 : Select .0 input for MUX10 01 : Select .1 input for MUX10 10 : Select .2 input for MUX10 11 : Select .3 input for MUX10 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-38. TRIP1MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	MUX9	R/W	0h	Select Bits for PWM-XBAR OUT1MUX9: 00 : Select .0 input for MUX9 01 : Select .1 input for MUX9 10 : Select .2 input for MUX9 11 : Select .3 input for MUX9 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for PWM-XBAR OUT1MUX8: 00 : Select .0 input for MUX8 01 : Select .1 input for MUX8 10 : Select .2 input for MUX8 11 : Select .3 input for MUX8 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for PWM-XBAR OUT1MUX7: 00 : Select .0 input for MUX7 01 : Select .1 input for MUX7 10 : Select .2 input for MUX7 11 : Select .3 input for MUX7 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13-12	MUX6	R/W	0h	Select Bits for PWM-XBAR OUT1MUX6: 00 : Select .0 input for MUX6 01 : Select .1 input for MUX6 10 : Select .2 input for MUX6 11 : Select .3 input for MUX6 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for PWM-XBAR OUT1MUX5: 00 : Select .0 input for MUX5 01 : Select .1 input for MUX5 10 : Select .2 input for MUX5 11 : Select .3 input for MUX5 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for PWM-XBAR OUT1MUX4: 00 : Select .0 input for MUX4 01 : Select .1 input for MUX4 10 : Select .2 input for MUX4 11 : Select .3 input for MUX4 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for PWM-XBAR OUT1MUX3: 00 : Select .0 input for MUX3 01 : Select .1 input for MUX3 10 : Select .2 input for MUX3 11 : Select .3 input for MUX3 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for PWM-XBAR OUT1MUX2: 00 : Select .0 input for MUX2 01 : Select .1 input for MUX2 10 : Select .2 input for MUX2 11 : Select .3 input for MUX2 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-38. TRIP1MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	MUX1	R/W	0h	Select Bits for PWM-XBAR OUT1MUX1: 00 : Select .0 input for MUX1 01 : Select .1 input for MUX1 10 : Select .2 input for MUX1 11 : Select .3 input for MUX1 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for PWM-XBAR OUT1MUX0: 00 : Select .0 input for MUX0 01 : Select .1 input for MUX0 10 : Select .2 input for MUX0 11 : Select .3 input for MUX0 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.2 TRIP1MUX16TO31CFG Register (Offset = 2h) [Reset = 00000000h]

TRIP1MUX16TO31CFG is shown in Figure 9-33 and described in Table 9-39.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output1

Figure 9-33. TRIP1MUX16TO31CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		MUX17		MUX16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-39. TRIP1MUX16TO31CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	MUX17	R/W	0h	Select Bits for PWM-XBAR OUT1MUX17: 00 : Select .0 input for MUX17 01 : Select .1 input for MUX17 10 : Select .2 input for MUX17 11 : Select .3 input for MUX17 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX16	R/W	0h	Select Bits for PWM-XBAR OUT1MUX16: 00 : Select .0 input for MUX16 01 : Select .1 input for MUX16 10 : Select .2 input for MUX16 11 : Select .3 input for MUX16 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.3 TRIP2MUX0TO15CFG Register (Offset = 4h) [Reset = 00000000h]

TRIP2MUX0TO15CFG is shown in Figure 9-34 and described in Table 9-40.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output2

Figure 9-34. TRIP2MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15		MUX14		MUX13		MUX12		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-40. TRIP2MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	MUX15	R/W	0h	Select Bits for PWM-XBAR OUT2MUX15: 00 : Select .0 input for MUX15 01 : Select .1 input for MUX15 10 : Select .2 input for MUX15 11 : Select .3 input for MUX15 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
29-28	MUX14	R/W	0h	Select Bits for PWM-XBAR OUT2MUX14: 00 : Select .0 input for MUX14 01 : Select .1 input for MUX14 10 : Select .2 input for MUX14 11 : Select .3 input for MUX14 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
27-26	MUX13	R/W	0h	Select Bits for PWM-XBAR OUT2MUX13: 00 : Select .0 input for MUX13 01 : Select .1 input for MUX13 10 : Select .2 input for MUX13 11 : Select .3 input for MUX13 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
25-24	MUX12	R/W	0h	Select Bits for PWM-XBAR OUT2MUX12: 00 : Select .0 input for MUX12 01 : Select .1 input for MUX12 10 : Select .2 input for MUX12 11 : Select .3 input for MUX12 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
23-22	MUX11	R/W	0h	Select Bits for PWM-XBAR OUT2MUX11: 00 : Select .0 input for MUX11 01 : Select .1 input for MUX11 10 : Select .2 input for MUX11 11 : Select .3 input for MUX11 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for PWM-XBAR OUT2MUX10: 00 : Select .0 input for MUX10 01 : Select .1 input for MUX10 10 : Select .2 input for MUX10 11 : Select .3 input for MUX10 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-40. TRIP2MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	MUX9	R/W	0h	Select Bits for PWM-XBAR OUT2MUX9: 00 : Select .0 input for MUX9 01 : Select .1 input for MUX9 10 : Select .2 input for MUX9 11 : Select .3 input for MUX9 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for PWM-XBAR OUT2MUX8: 00 : Select .0 input for MUX8 01 : Select .1 input for MUX8 10 : Select .2 input for MUX8 11 : Select .3 input for MUX8 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for PWM-XBAR OUT2MUX7: 00 : Select .0 input for MUX7 01 : Select .1 input for MUX7 10 : Select .2 input for MUX7 11 : Select .3 input for MUX7 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13-12	MUX6	R/W	0h	Select Bits for PWM-XBAR OUT2MUX6: 00 : Select .0 input for MUX6 01 : Select .1 input for MUX6 10 : Select .2 input for MUX6 11 : Select .3 input for MUX6 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for PWM-XBAR OUT2MUX5: 00 : Select .0 input for MUX5 01 : Select .1 input for MUX5 10 : Select .2 input for MUX5 11 : Select .3 input for MUX5 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for PWM-XBAR OUT2MUX4: 00 : Select .0 input for MUX4 01 : Select .1 input for MUX4 10 : Select .2 input for MUX4 11 : Select .3 input for MUX4 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for PWM-XBAR OUT2MUX3: 00 : Select .0 input for MUX3 01 : Select .1 input for MUX3 10 : Select .2 input for MUX3 11 : Select .3 input for MUX3 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for PWM-XBAR OUT2MUX2: 00 : Select .0 input for MUX2 01 : Select .1 input for MUX2 10 : Select .2 input for MUX2 11 : Select .3 input for MUX2 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-40. TRIP2MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	MUX1	R/W	0h	Select Bits for PWM-XBAR OUT2MUX1: 00 : Select .0 input for MUX1 01 : Select .1 input for MUX1 10 : Select .2 input for MUX1 11 : Select .3 input for MUX1 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for PWM-XBAR OUT2MUX0: 00 : Select .0 input for MUX0 01 : Select .1 input for MUX0 10 : Select .2 input for MUX0 11 : Select .3 input for MUX0 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.4 TRIP2MUX16TO31CFG Register (Offset = 6h) [Reset = 00000000h]

TRIP2MUX16TO31CFG is shown in Figure 9-35 and described in Table 9-41.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output2

Figure 9-35. TRIP2MUX16TO31CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		MUX17		MUX16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-41. TRIP2MUX16TO31CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	MUX17	R/W	0h	Select Bits for PWM-XBAR OUT2MUX17: 00 : Select .0 input for MUX17 01 : Select .1 input for MUX17 10 : Select .2 input for MUX17 11 : Select .3 input for MUX17 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX16	R/W	0h	Select Bits for PWM-XBAR OUT2MUX16: 00 : Select .0 input for MUX16 01 : Select .1 input for MUX16 10 : Select .2 input for MUX16 11 : Select .3 input for MUX16 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.5 TRIP3MUX0TO15CFG Register (Offset = 8h) [Reset = 00000000h]

TRIP3MUX0TO15CFG is shown in Figure 9-36 and described in Table 9-42.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output3

Figure 9-36. TRIP3MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15		MUX14		MUX13		MUX12		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-42. TRIP3MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	MUX15	R/W	0h	Select Bits for PWM-XBAR OUT3MUX15: 00 : Select .0 input for MUX15 01 : Select .1 input for MUX15 10 : Select .2 input for MUX15 11 : Select .3 input for MUX15 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
29-28	MUX14	R/W	0h	Select Bits for PWM-XBAR OUT3MUX14: 00 : Select .0 input for MUX14 01 : Select .1 input for MUX14 10 : Select .2 input for MUX14 11 : Select .3 input for MUX14 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
27-26	MUX13	R/W	0h	Select Bits for PWM-XBAR OUT3MUX13: 00 : Select .0 input for MUX13 01 : Select .1 input for MUX13 10 : Select .2 input for MUX13 11 : Select .3 input for MUX13 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
25-24	MUX12	R/W	0h	Select Bits for PWM-XBAR OUT3MUX12: 00 : Select .0 input for MUX12 01 : Select .1 input for MUX12 10 : Select .2 input for MUX12 11 : Select .3 input for MUX12 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
23-22	MUX11	R/W	0h	Select Bits for PWM-XBAR OUT3MUX11: 00 : Select .0 input for MUX11 01 : Select .1 input for MUX11 10 : Select .2 input for MUX11 11 : Select .3 input for MUX11 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for PWM-XBAR OUT3MUX10: 00 : Select .0 input for MUX10 01 : Select .1 input for MUX10 10 : Select .2 input for MUX10 11 : Select .3 input for MUX10 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-42. TRIP3MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	MUX9	R/W	0h	Select Bits for PWM-XBAR OUT3MUX9: 00 : Select .0 input for MUX9 01 : Select .1 input for MUX9 10 : Select .2 input for MUX9 11 : Select .3 input for MUX9 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for PWM-XBAR OUT3MUX8: 00 : Select .0 input for MUX8 01 : Select .1 input for MUX8 10 : Select .2 input for MUX8 11 : Select .3 input for MUX8 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for PWM-XBAR OUT3MUX7: 00 : Select .0 input for MUX7 01 : Select .1 input for MUX7 10 : Select .2 input for MUX7 11 : Select .3 input for MUX7 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13-12	MUX6	R/W	0h	Select Bits for PWM-XBAR OUT3MUX6: 00 : Select .0 input for MUX6 01 : Select .1 input for MUX6 10 : Select .2 input for MUX6 11 : Select .3 input for MUX6 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for PWM-XBAR OUT3MUX5: 00 : Select .0 input for MUX5 01 : Select .1 input for MUX5 10 : Select .2 input for MUX5 11 : Select .3 input for MUX5 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for PWM-XBAR OUT3MUX4: 00 : Select .0 input for MUX4 01 : Select .1 input for MUX4 10 : Select .2 input for MUX4 11 : Select .3 input for MUX4 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for PWM-XBAR OUT3MUX3: 00 : Select .0 input for MUX3 01 : Select .1 input for MUX3 10 : Select .2 input for MUX3 11 : Select .3 input for MUX3 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for PWM-XBAR OUT3MUX2: 00 : Select .0 input for MUX2 01 : Select .1 input for MUX2 10 : Select .2 input for MUX2 11 : Select .3 input for MUX2 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-42. TRIP3MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	MUX1	R/W	0h	Select Bits for PWM-XBAR OUT3MUX1: 00 : Select .0 input for MUX1 01 : Select .1 input for MUX1 10 : Select .2 input for MUX1 11 : Select .3 input for MUX1 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for PWM-XBAR OUT3MUX0: 00 : Select .0 input for MUX0 01 : Select .1 input for MUX0 10 : Select .2 input for MUX0 11 : Select .3 input for MUX0 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.6 TRIP3MUX16TO31CFG Register (Offset = Ah) [Reset = 00000000h]

TRIP3MUX16TO31CFG is shown in Figure 9-37 and described in Table 9-43.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output3

Figure 9-37. TRIP3MUX16TO31CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		MUX17		MUX16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-43. TRIP3MUX16TO31CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	MUX17	R/W	0h	Select Bits for PWM-XBAR OUT3MUX17: 00 : Select .0 input for MUX17 01 : Select .1 input for MUX17 10 : Select .2 input for MUX17 11 : Select .3 input for MUX17 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX16	R/W	0h	Select Bits for PWM-XBAR OUT3MUX16: 00 : Select .0 input for MUX16 01 : Select .1 input for MUX16 10 : Select .2 input for MUX16 11 : Select .3 input for MUX16 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.7 TRIP4MUX0TO15CFG Register (Offset = Ch) [Reset = 00000000h]

TRIP4MUX0TO15CFG is shown in Figure 9-38 and described in Table 9-44.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output4

Figure 9-38. TRIP4MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15		MUX14		MUX13		MUX12		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-44. TRIP4MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	MUX15	R/W	0h	Select Bits for PWM-XBAR OUT4MUX15: 00 : Select .0 input for MUX15 01 : Select .1 input for MUX15 10 : Select .2 input for MUX15 11 : Select .3 input for MUX15 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
29-28	MUX14	R/W	0h	Select Bits for PWM-XBAR OUT4MUX14: 00 : Select .0 input for MUX14 01 : Select .1 input for MUX14 10 : Select .2 input for MUX14 11 : Select .3 input for MUX14 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
27-26	MUX13	R/W	0h	Select Bits for PWM-XBAR OUT4MUX13: 00 : Select .0 input for MUX13 01 : Select .1 input for MUX13 10 : Select .2 input for MUX13 11 : Select .3 input for MUX13 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
25-24	MUX12	R/W	0h	Select Bits for PWM-XBAR OUT4MUX12: 00 : Select .0 input for MUX12 01 : Select .1 input for MUX12 10 : Select .2 input for MUX12 11 : Select .3 input for MUX12 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
23-22	MUX11	R/W	0h	Select Bits for PWM-XBAR OUT4MUX11: 00 : Select .0 input for MUX11 01 : Select .1 input for MUX11 10 : Select .2 input for MUX11 11 : Select .3 input for MUX11 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for PWM-XBAR OUT4MUX10: 00 : Select .0 input for MUX10 01 : Select .1 input for MUX10 10 : Select .2 input for MUX10 11 : Select .3 input for MUX10 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-44. TRIP4MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	MUX9	R/W	0h	Select Bits for PWM-XBAR OUT4MUX9: 00 : Select .0 input for MUX9 01 : Select .1 input for MUX9 10 : Select .2 input for MUX9 11 : Select .3 input for MUX9 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for PWM-XBAR OUT4MUX8: 00 : Select .0 input for MUX8 01 : Select .1 input for MUX8 10 : Select .2 input for MUX8 11 : Select .3 input for MUX8 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for PWM-XBAR OUT4MUX7: 00 : Select .0 input for MUX7 01 : Select .1 input for MUX7 10 : Select .2 input for MUX7 11 : Select .3 input for MUX7 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13-12	MUX6	R/W	0h	Select Bits for PWM-XBAR OUT4MUX6: 00 : Select .0 input for MUX6 01 : Select .1 input for MUX6 10 : Select .2 input for MUX6 11 : Select .3 input for MUX6 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for PWM-XBAR OUT4MUX5: 00 : Select .0 input for MUX5 01 : Select .1 input for MUX5 10 : Select .2 input for MUX5 11 : Select .3 input for MUX5 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for PWM-XBAR OUT4MUX4: 00 : Select .0 input for MUX4 01 : Select .1 input for MUX4 10 : Select .2 input for MUX4 11 : Select .3 input for MUX4 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for PWM-XBAR OUT4MUX3: 00 : Select .0 input for MUX3 01 : Select .1 input for MUX3 10 : Select .2 input for MUX3 11 : Select .3 input for MUX3 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for PWM-XBAR OUT4MUX2: 00 : Select .0 input for MUX2 01 : Select .1 input for MUX2 10 : Select .2 input for MUX2 11 : Select .3 input for MUX2 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-44. TRIP4MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	MUX1	R/W	0h	Select Bits for PWM-XBAR OUT4MUX1: 00 : Select .0 input for MUX1 01 : Select .1 input for MUX1 10 : Select .2 input for MUX1 11 : Select .3 input for MUX1 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for PWM-XBAR OUT4MUX0: 00 : Select .0 input for MUX0 01 : Select .1 input for MUX0 10 : Select .2 input for MUX0 11 : Select .3 input for MUX0 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.8 TRIP4MUX16TO31CFG Register (Offset = Eh) [Reset = 00000000h]

TRIP4MUX16TO31CFG is shown in Figure 9-39 and described in Table 9-45.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output4

Figure 9-39. TRIP4MUX16TO31CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		MUX17		MUX16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-45. TRIP4MUX16TO31CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	MUX17	R/W	0h	Select Bits for PWM-XBAR OUT4MUX17: 00 : Select .0 input for MUX17 01 : Select .1 input for MUX17 10 : Select .2 input for MUX17 11 : Select .3 input for MUX17 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX16	R/W	0h	Select Bits for PWM-XBAR OUT4MUX16: 00 : Select .0 input for MUX16 01 : Select .1 input for MUX16 10 : Select .2 input for MUX16 11 : Select .3 input for MUX16 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.9 TRIP5MUX0TO15CFG Register (Offset = 10h) [Reset = 00000000h]

TRIP5MUX0TO15CFG is shown in Figure 9-40 and described in Table 9-46.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output5

Figure 9-40. TRIP5MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15		MUX14		MUX13		MUX12		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-46. TRIP5MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	MUX15	R/W	0h	Select Bits for PWM-XBAR OUT5MUX15: 00 : Select .0 input for MUX15 01 : Select .1 input for MUX15 10 : Select .2 input for MUX15 11 : Select .3 input for MUX15 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
29-28	MUX14	R/W	0h	Select Bits for PWM-XBAR OUT5MUX14: 00 : Select .0 input for MUX14 01 : Select .1 input for MUX14 10 : Select .2 input for MUX14 11 : Select .3 input for MUX14 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
27-26	MUX13	R/W	0h	Select Bits for PWM-XBAR OUT5MUX13: 00 : Select .0 input for MUX13 01 : Select .1 input for MUX13 10 : Select .2 input for MUX13 11 : Select .3 input for MUX13 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
25-24	MUX12	R/W	0h	Select Bits for PWM-XBAR OUT5MUX12: 00 : Select .0 input for MUX12 01 : Select .1 input for MUX12 10 : Select .2 input for MUX12 11 : Select .3 input for MUX12 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
23-22	MUX11	R/W	0h	Select Bits for PWM-XBAR OUT5MUX11: 00 : Select .0 input for MUX11 01 : Select .1 input for MUX11 10 : Select .2 input for MUX11 11 : Select .3 input for MUX11 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for PWM-XBAR OUT5MUX10: 00 : Select .0 input for MUX10 01 : Select .1 input for MUX10 10 : Select .2 input for MUX10 11 : Select .3 input for MUX10 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-46. TRIP5MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	MUX9	R/W	0h	Select Bits for PWM-XBAR OUT5MUX9: 00 : Select .0 input for MUX9 01 : Select .1 input for MUX9 10 : Select .2 input for MUX9 11 : Select .3 input for MUX9 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for PWM-XBAR OUT5MUX8: 00 : Select .0 input for MUX8 01 : Select .1 input for MUX8 10 : Select .2 input for MUX8 11 : Select .3 input for MUX8 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for PWM-XBAR OUT5MUX7: 00 : Select .0 input for MUX7 01 : Select .1 input for MUX7 10 : Select .2 input for MUX7 11 : Select .3 input for MUX7 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13-12	MUX6	R/W	0h	Select Bits for PWM-XBAR OUT5MUX6: 00 : Select .0 input for MUX6 01 : Select .1 input for MUX6 10 : Select .2 input for MUX6 11 : Select .3 input for MUX6 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for PWM-XBAR OUT5MUX5: 00 : Select .0 input for MUX5 01 : Select .1 input for MUX5 10 : Select .2 input for MUX5 11 : Select .3 input for MUX5 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for PWM-XBAR OUT5MUX4: 00 : Select .0 input for MUX4 01 : Select .1 input for MUX4 10 : Select .2 input for MUX4 11 : Select .3 input for MUX4 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for PWM-XBAR OUT5MUX3: 00 : Select .0 input for MUX3 01 : Select .1 input for MUX3 10 : Select .2 input for MUX3 11 : Select .3 input for MUX3 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for PWM-XBAR OUT5MUX2: 00 : Select .0 input for MUX2 01 : Select .1 input for MUX2 10 : Select .2 input for MUX2 11 : Select .3 input for MUX2 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-46. TRIP5MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	MUX1	R/W	0h	Select Bits for PWM-XBAR OUT5MUX1: 00 : Select .0 input for MUX1 01 : Select .1 input for MUX1 10 : Select .2 input for MUX1 11 : Select .3 input for MUX1 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for PWM-XBAR OUT5MUX0: 00 : Select .0 input for MUX0 01 : Select .1 input for MUX0 10 : Select .2 input for MUX0 11 : Select .3 input for MUX0 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.10 TRIP5MUX16TO31CFG Register (Offset = 12h) [Reset = 00000000h]

TRIP5MUX16TO31CFG is shown in Figure 9-41 and described in Table 9-47.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output5

Figure 9-41. TRIP5MUX16TO31CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		MUX17		MUX16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-47. TRIP5MUX16TO31CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	MUX17	R/W	0h	Select Bits for PWM-XBAR OUT5MUX17: 00 : Select .0 input for MUX17 01 : Select .1 input for MUX17 10 : Select .2 input for MUX17 11 : Select .3 input for MUX17 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX16	R/W	0h	Select Bits for PWM-XBAR OUT5MUX16: 00 : Select .0 input for MUX16 01 : Select .1 input for MUX16 10 : Select .2 input for MUX16 11 : Select .3 input for MUX16 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.11 TRIP6MUX0TO15CFG Register (Offset = 14h) [Reset = 00000000h]

TRIP6MUX0TO15CFG is shown in Figure 9-42 and described in Table 9-48.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output6

Figure 9-42. TRIP6MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15		MUX14		MUX13		MUX12		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-48. TRIP6MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	MUX15	R/W	0h	Select Bits for PWM-XBAR OUT6MUX15: 00 : Select .0 input for MUX15 01 : Select .1 input for MUX15 10 : Select .2 input for MUX15 11 : Select .3 input for MUX15 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
29-28	MUX14	R/W	0h	Select Bits for PWM-XBAR OUT6MUX14: 00 : Select .0 input for MUX14 01 : Select .1 input for MUX14 10 : Select .2 input for MUX14 11 : Select .3 input for MUX14 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
27-26	MUX13	R/W	0h	Select Bits for PWM-XBAR OUT6MUX13: 00 : Select .0 input for MUX13 01 : Select .1 input for MUX13 10 : Select .2 input for MUX13 11 : Select .3 input for MUX13 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
25-24	MUX12	R/W	0h	Select Bits for PWM-XBAR OUT6MUX12: 00 : Select .0 input for MUX12 01 : Select .1 input for MUX12 10 : Select .2 input for MUX12 11 : Select .3 input for MUX12 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
23-22	MUX11	R/W	0h	Select Bits for PWM-XBAR OUT6MUX11: 00 : Select .0 input for MUX11 01 : Select .1 input for MUX11 10 : Select .2 input for MUX11 11 : Select .3 input for MUX11 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for PWM-XBAR OUT6MUX10: 00 : Select .0 input for MUX10 01 : Select .1 input for MUX10 10 : Select .2 input for MUX10 11 : Select .3 input for MUX10 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-48. TRIP6MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	MUX9	R/W	0h	Select Bits for PWM-XBAR OUT6MUX9: 00 : Select .0 input for MUX9 01 : Select .1 input for MUX9 10 : Select .2 input for MUX9 11 : Select .3 input for MUX9 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for PWM-XBAR OUT6MUX8: 00 : Select .0 input for MUX8 01 : Select .1 input for MUX8 10 : Select .2 input for MUX8 11 : Select .3 input for MUX8 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for PWM-XBAR OUT6MUX7: 00 : Select .0 input for MUX7 01 : Select .1 input for MUX7 10 : Select .2 input for MUX7 11 : Select .3 input for MUX7 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13-12	MUX6	R/W	0h	Select Bits for PWM-XBAR OUT6MUX6: 00 : Select .0 input for MUX6 01 : Select .1 input for MUX6 10 : Select .2 input for MUX6 11 : Select .3 input for MUX6 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for PWM-XBAR OUT6MUX5: 00 : Select .0 input for MUX5 01 : Select .1 input for MUX5 10 : Select .2 input for MUX5 11 : Select .3 input for MUX5 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for PWM-XBAR OUT6MUX4: 00 : Select .0 input for MUX4 01 : Select .1 input for MUX4 10 : Select .2 input for MUX4 11 : Select .3 input for MUX4 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for PWM-XBAR OUT6MUX3: 00 : Select .0 input for MUX3 01 : Select .1 input for MUX3 10 : Select .2 input for MUX3 11 : Select .3 input for MUX3 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for PWM-XBAR OUT6MUX2: 00 : Select .0 input for MUX2 01 : Select .1 input for MUX2 10 : Select .2 input for MUX2 11 : Select .3 input for MUX2 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-48. TRIP6MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	MUX1	R/W	0h	Select Bits for PWM-XBAR OUT6MUX1: 00 : Select .0 input for MUX1 01 : Select .1 input for MUX1 10 : Select .2 input for MUX1 11 : Select .3 input for MUX1 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for PWM-XBAR OUT6MUX0: 00 : Select .0 input for MUX0 01 : Select .1 input for MUX0 10 : Select .2 input for MUX0 11 : Select .3 input for MUX0 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.12 TRIP6MUX16TO31CFG Register (Offset = 16h) [Reset = 00000000h]

TRIP6MUX16TO31CFG is shown in Figure 9-43 and described in Table 9-49.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output6

Figure 9-43. TRIP6MUX16TO31CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		MUX17		MUX16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-49. TRIP6MUX16TO31CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	MUX17	R/W	0h	Select Bits for PWM-XBAR OUT6MUX17: 00 : Select .0 input for MUX17 01 : Select .1 input for MUX17 10 : Select .2 input for MUX17 11 : Select .3 input for MUX17 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX16	R/W	0h	Select Bits for PWM-XBAR OUT6MUX16: 00 : Select .0 input for MUX16 01 : Select .1 input for MUX16 10 : Select .2 input for MUX16 11 : Select .3 input for MUX16 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.13 TRIP7MUX0TO15CFG Register (Offset = 18h) [Reset = 00000000h]

TRIP7MUX0TO15CFG is shown in Figure 9-44 and described in Table 9-50.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output7

Figure 9-44. TRIP7MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15		MUX14		MUX13		MUX12		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-50. TRIP7MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	MUX15	R/W	0h	Select Bits for PWM-XBAR OUT7MUX15: 00 : Select .0 input for MUX15 01 : Select .1 input for MUX15 10 : Select .2 input for MUX15 11 : Select .3 input for MUX15 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
29-28	MUX14	R/W	0h	Select Bits for PWM-XBAR OUT7MUX14: 00 : Select .0 input for MUX14 01 : Select .1 input for MUX14 10 : Select .2 input for MUX14 11 : Select .3 input for MUX14 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
27-26	MUX13	R/W	0h	Select Bits for PWM-XBAR OUT7MUX13: 00 : Select .0 input for MUX13 01 : Select .1 input for MUX13 10 : Select .2 input for MUX13 11 : Select .3 input for MUX13 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
25-24	MUX12	R/W	0h	Select Bits for PWM-XBAR OUT7MUX12: 00 : Select .0 input for MUX12 01 : Select .1 input for MUX12 10 : Select .2 input for MUX12 11 : Select .3 input for MUX12 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
23-22	MUX11	R/W	0h	Select Bits for PWM-XBAR OUT7MUX11: 00 : Select .0 input for MUX11 01 : Select .1 input for MUX11 10 : Select .2 input for MUX11 11 : Select .3 input for MUX11 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for PWM-XBAR OUT7MUX10: 00 : Select .0 input for MUX10 01 : Select .1 input for MUX10 10 : Select .2 input for MUX10 11 : Select .3 input for MUX10 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-50. TRIP7MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	MUX9	R/W	0h	Select Bits for PWM-XBAR OUT7MUX9: 00 : Select .0 input for MUX9 01 : Select .1 input for MUX9 10 : Select .2 input for MUX9 11 : Select .3 input for MUX9 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for PWM-XBAR OUT7MUX8: 00 : Select .0 input for MUX8 01 : Select .1 input for MUX8 10 : Select .2 input for MUX8 11 : Select .3 input for MUX8 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for PWM-XBAR OUT7MUX7: 00 : Select .0 input for MUX7 01 : Select .1 input for MUX7 10 : Select .2 input for MUX7 11 : Select .3 input for MUX7 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13-12	MUX6	R/W	0h	Select Bits for PWM-XBAR OUT7MUX6: 00 : Select .0 input for MUX6 01 : Select .1 input for MUX6 10 : Select .2 input for MUX6 11 : Select .3 input for MUX6 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for PWM-XBAR OUT7MUX5: 00 : Select .0 input for MUX5 01 : Select .1 input for MUX5 10 : Select .2 input for MUX5 11 : Select .3 input for MUX5 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for PWM-XBAR OUT7MUX4: 00 : Select .0 input for MUX4 01 : Select .1 input for MUX4 10 : Select .2 input for MUX4 11 : Select .3 input for MUX4 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for PWM-XBAR OUT7MUX3: 00 : Select .0 input for MUX3 01 : Select .1 input for MUX3 10 : Select .2 input for MUX3 11 : Select .3 input for MUX3 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for PWM-XBAR OUT7MUX2: 00 : Select .0 input for MUX2 01 : Select .1 input for MUX2 10 : Select .2 input for MUX2 11 : Select .3 input for MUX2 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-50. TRIP7MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	MUX1	R/W	0h	Select Bits for PWM-XBAR OUT7MUX1: 00 : Select .0 input for MUX1 01 : Select .1 input for MUX1 10 : Select .2 input for MUX1 11 : Select .3 input for MUX1 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for PWM-XBAR OUT7MUX0: 00 : Select .0 input for MUX0 01 : Select .1 input for MUX0 10 : Select .2 input for MUX0 11 : Select .3 input for MUX0 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.14 TRIP7MUX16TO31CFG Register (Offset = 1Ah) [Reset = 00000000h]

TRIP7MUX16TO31CFG is shown in Figure 9-45 and described in Table 9-51.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output7

Figure 9-45. TRIP7MUX16TO31CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		MUX17		MUX16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-51. TRIP7MUX16TO31CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	MUX17	R/W	0h	Select Bits for PWM-XBAR OUT7MUX17: 00 : Select .0 input for MUX17 01 : Select .1 input for MUX17 10 : Select .2 input for MUX17 11 : Select .3 input for MUX17 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX16	R/W	0h	Select Bits for PWM-XBAR OUT7MUX16: 00 : Select .0 input for MUX16 01 : Select .1 input for MUX16 10 : Select .2 input for MUX16 11 : Select .3 input for MUX16 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.15 TRIP8MUX0TO15CFG Register (Offset = 1Ch) [Reset = 00000000h]

TRIP8MUX0TO15CFG is shown in Figure 9-46 and described in Table 9-52.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output8

Figure 9-46. TRIP8MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX15		MUX14		MUX13		MUX12		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-52. TRIP8MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	MUX15	R/W	0h	Select Bits for PWM-XBAR OUT8MUX15: 00 : Select .0 input for MUX15 01 : Select .1 input for MUX15 10 : Select .2 input for MUX15 11 : Select .3 input for MUX15 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
29-28	MUX14	R/W	0h	Select Bits for PWM-XBAR OUT8MUX14: 00 : Select .0 input for MUX14 01 : Select .1 input for MUX14 10 : Select .2 input for MUX14 11 : Select .3 input for MUX14 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
27-26	MUX13	R/W	0h	Select Bits for PWM-XBAR OUT8MUX13: 00 : Select .0 input for MUX13 01 : Select .1 input for MUX13 10 : Select .2 input for MUX13 11 : Select .3 input for MUX13 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
25-24	MUX12	R/W	0h	Select Bits for PWM-XBAR OUT8MUX12: 00 : Select .0 input for MUX12 01 : Select .1 input for MUX12 10 : Select .2 input for MUX12 11 : Select .3 input for MUX12 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
23-22	MUX11	R/W	0h	Select Bits for PWM-XBAR OUT8MUX11: 00 : Select .0 input for MUX11 01 : Select .1 input for MUX11 10 : Select .2 input for MUX11 11 : Select .3 input for MUX11 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for PWM-XBAR OUT8MUX10: 00 : Select .0 input for MUX10 01 : Select .1 input for MUX10 10 : Select .2 input for MUX10 11 : Select .3 input for MUX10 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-52. TRIP8MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	MUX9	R/W	0h	Select Bits for PWM-XBAR OUT8MUX9: 00 : Select .0 input for MUX9 01 : Select .1 input for MUX9 10 : Select .2 input for MUX9 11 : Select .3 input for MUX9 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for PWM-XBAR OUT8MUX8: 00 : Select .0 input for MUX8 01 : Select .1 input for MUX8 10 : Select .2 input for MUX8 11 : Select .3 input for MUX8 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for PWM-XBAR OUT8MUX7: 00 : Select .0 input for MUX7 01 : Select .1 input for MUX7 10 : Select .2 input for MUX7 11 : Select .3 input for MUX7 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13-12	MUX6	R/W	0h	Select Bits for PWM-XBAR OUT8MUX6: 00 : Select .0 input for MUX6 01 : Select .1 input for MUX6 10 : Select .2 input for MUX6 11 : Select .3 input for MUX6 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for PWM-XBAR OUT8MUX5: 00 : Select .0 input for MUX5 01 : Select .1 input for MUX5 10 : Select .2 input for MUX5 11 : Select .3 input for MUX5 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for PWM-XBAR OUT8MUX4: 00 : Select .0 input for MUX4 01 : Select .1 input for MUX4 10 : Select .2 input for MUX4 11 : Select .3 input for MUX4 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for PWM-XBAR OUT8MUX3: 00 : Select .0 input for MUX3 01 : Select .1 input for MUX3 10 : Select .2 input for MUX3 11 : Select .3 input for MUX3 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for PWM-XBAR OUT8MUX2: 00 : Select .0 input for MUX2 01 : Select .1 input for MUX2 10 : Select .2 input for MUX2 11 : Select .3 input for MUX2 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-52. TRIP8MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	MUX1	R/W	0h	Select Bits for PWM-XBAR OUT8MUX1: 00 : Select .0 input for MUX1 01 : Select .1 input for MUX1 10 : Select .2 input for MUX1 11 : Select .3 input for MUX1 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for PWM-XBAR OUT8MUX0: 00 : Select .0 input for MUX0 01 : Select .1 input for MUX0 10 : Select .2 input for MUX0 11 : Select .3 input for MUX0 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.16 TRIP8MUX16TO31CFG Register (Offset = 1Eh) [Reset = 00000000h]

TRIP8MUX16TO31CFG is shown in Figure 9-47 and described in Table 9-53.

Return to the Summary Table.

PWM XBAR Mux Configuration for Output8

Figure 9-47. TRIP8MUX16TO31CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		RESERVED		MUX17		MUX16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-53. TRIP8MUX16TO31CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	0h	Reserved
21-20	RESERVED	R/W	0h	Reserved
19-18	RESERVED	R/W	0h	Reserved
17-16	RESERVED	R/W	0h	Reserved
15-14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R/W	0h	Reserved
11-10	RESERVED	R/W	0h	Reserved
9-8	RESERVED	R/W	0h	Reserved
7-6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R/W	0h	Reserved
3-2	MUX17	R/W	0h	Select Bits for PWM-XBAR OUT8MUX17: 00 : Select .0 input for MUX17 01 : Select .1 input for MUX17 10 : Select .2 input for MUX17 11 : Select .3 input for MUX17 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX16	R/W	0h	Select Bits for PWM-XBAR OUT8MUX16: 00 : Select .0 input for MUX16 01 : Select .1 input for MUX16 10 : Select .2 input for MUX16 11 : Select .3 input for MUX16 Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.17 TRIP1MUXENABLE Register (Offset = 20h) [Reset = 00000000h]

TRIP1MUXENABLE is shown in Figure 9-48 and described in Table 9-54.

Return to the Summary Table.

PWM XBAR Mux Enable for Output1

Figure 9-48. TRIP1MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	MUX17	MUX16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
MUX15	MUX14	MUX13	MUX12	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-54. TRIP1MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	MUX17	R/W	0h	Selects the output of MUX17 to drive OUT1 of PWM-XBAR 0: Respective output of MUX17 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX17 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
16	MUX16	R/W	0h	Selects the output of MUX16 to drive OUT1 of PWM-XBAR 0: Respective output of MUX16 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX16 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-54. TRIP1MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	MUX15	R/W	0h	Selects the output of MUX15 to drive OUT1 of PWM-XBAR 0: Respective output of MUX15 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX15 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
14	MUX14	R/W	0h	Selects the output of MUX14 to drive OUT1 of PWM-XBAR 0: Respective output of MUX14 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX14 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13	MUX13	R/W	0h	Selects the output of MUX13 to drive OUT1 of PWM-XBAR 0: Respective output of MUX13 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX13 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
12	MUX12	R/W	0h	Selects the output of MUX12 to drive OUT1 of PWM-XBAR 0: Respective output of MUX12 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX12 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11	MUX11	R/W	0h	Selects the output of MUX11 to drive OUT1 of PWM-XBAR 0: Respective output of MUX11 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX11 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
10	MUX10	R/W	0h	Selects the output of MUX10 to drive OUT1 of PWM-XBAR 0: Respective output of MUX10 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX10 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of MUX9 to drive OUT1 of PWM-XBAR 0: Respective output of MUX9 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX9 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of MUX8 to drive OUT1 of PWM-XBAR 0: Respective output of MUX8 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX8 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-54. TRIP1MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	MUX7	R/W	0h	Selects the output of MUX7 to drive OUT1 of PWM-XBAR 0: Respective output of MUX7 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX7 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of MUX6 to drive OUT1 of PWM-XBAR 0: Respective output of MUX6 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX6 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of MUX5 to drive OUT1 of PWM-XBAR 0: Respective output of MUX5 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX5 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of MUX4 to drive OUT1 of PWM-XBAR 0: Respective output of MUX4 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX4 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of MUX3 to drive OUT1 of PWM-XBAR 0: Respective output of MUX3 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX3 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	MUX2	R/W	0h	Selects the output of MUX2 to drive OUT1 of PWM-XBAR 0: Respective output of MUX2 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX2 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of MUX1 to drive OUT1 of PWM-XBAR 0: Respective output of MUX1 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX1 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of MUX0 to drive OUT1 of PWM-XBAR 0: Respective output of MUX0 is disabled to drive the OUT1 of PWM-XBAR 1: Respective output of MUX0 is enabled to drive the OUT1 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.18 TRIP2MUXENABLE Register (Offset = 22h) [Reset = 00000000h]

TRIP2MUXENABLE is shown in Figure 9-49 and described in Table 9-55.

Return to the Summary Table.

PWM XBAR Mux Enable for Output2

Figure 9-49. TRIP2MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	MUX17	MUX16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
MUX15	MUX14	MUX13	MUX12	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-55. TRIP2MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	MUX17	R/W	0h	Selects the output of MUX17 to drive OUT2 of PWM-XBAR 0: Respective output of MUX17 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX17 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
16	MUX16	R/W	0h	Selects the output of MUX16 to drive OUT2 of PWM-XBAR 0: Respective output of MUX16 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX16 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-55. TRIP2MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	MUX15	R/W	0h	Selects the output of MUX15 to drive OUT2 of PWM-XBAR 0: Respective output of MUX15 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX15 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
14	MUX14	R/W	0h	Selects the output of MUX14 to drive OUT2 of PWM-XBAR 0: Respective output of MUX14 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX14 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13	MUX13	R/W	0h	Selects the output of MUX13 to drive OUT2 of PWM-XBAR 0: Respective output of MUX13 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX13 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
12	MUX12	R/W	0h	Selects the output of MUX12 to drive OUT2 of PWM-XBAR 0: Respective output of MUX12 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX12 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11	MUX11	R/W	0h	Selects the output of MUX11 to drive OUT2 of PWM-XBAR 0: Respective output of MUX11 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX11 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
10	MUX10	R/W	0h	Selects the output of MUX10 to drive OUT2 of PWM-XBAR 0: Respective output of MUX10 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX10 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of MUX9 to drive OUT2 of PWM-XBAR 0: Respective output of MUX9 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX9 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of MUX8 to drive OUT2 of PWM-XBAR 0: Respective output of MUX8 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX8 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-55. TRIP2MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	MUX7	R/W	0h	Selects the output of MUX7 to drive OUT2 of PWM-XBAR 0: Respective output of MUX7 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX7 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of MUX6 to drive OUT2 of PWM-XBAR 0: Respective output of MUX6 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX6 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of MUX5 to drive OUT2 of PWM-XBAR 0: Respective output of MUX5 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX5 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of MUX4 to drive OUT2 of PWM-XBAR 0: Respective output of MUX4 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX4 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of MUX3 to drive OUT2 of PWM-XBAR 0: Respective output of MUX3 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX3 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	MUX2	R/W	0h	Selects the output of MUX2 to drive OUT2 of PWM-XBAR 0: Respective output of MUX2 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX2 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of MUX1 to drive OUT2 of PWM-XBAR 0: Respective output of MUX1 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX1 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of MUX0 to drive OUT2 of PWM-XBAR 0: Respective output of MUX0 is disabled to drive the OUT2 of PWM-XBAR 1: Respective output of MUX0 is enabled to drive the OUT2 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.19 TRIP3MUXENABLE Register (Offset = 24h) [Reset = 00000000h]

TRIP3MUXENABLE is shown in Figure 9-50 and described in Table 9-56.

Return to the Summary Table.

PWM XBAR Mux Enable for Output3

Figure 9-50. TRIP3MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	MUX17	MUX16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
MUX15	MUX14	MUX13	MUX12	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-56. TRIP3MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	MUX17	R/W	0h	Selects the output of MUX17 to drive OUT3 of PWM-XBAR 0: Respective output of MUX17 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX17 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
16	MUX16	R/W	0h	Selects the output of MUX16 to drive OUT3 of PWM-XBAR 0: Respective output of MUX16 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX16 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-56. TRIP3MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	MUX15	R/W	0h	Selects the output of MUX15 to drive OUT3 of PWM-XBAR 0: Respective output of MUX15 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX15 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
14	MUX14	R/W	0h	Selects the output of MUX14 to drive OUT3 of PWM-XBAR 0: Respective output of MUX14 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX14 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13	MUX13	R/W	0h	Selects the output of MUX13 to drive OUT3 of PWM-XBAR 0: Respective output of MUX13 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX13 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
12	MUX12	R/W	0h	Selects the output of MUX12 to drive OUT3 of PWM-XBAR 0: Respective output of MUX12 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX12 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11	MUX11	R/W	0h	Selects the output of MUX11 to drive OUT3 of PWM-XBAR 0: Respective output of MUX11 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX11 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
10	MUX10	R/W	0h	Selects the output of MUX10 to drive OUT3 of PWM-XBAR 0: Respective output of MUX10 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX10 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of MUX9 to drive OUT3 of PWM-XBAR 0: Respective output of MUX9 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX9 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of MUX8 to drive OUT3 of PWM-XBAR 0: Respective output of MUX8 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX8 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-56. TRIP3MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	MUX7	R/W	0h	Selects the output of MUX7 to drive OUT3 of PWM-XBAR 0: Respective output of MUX7 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX7 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of MUX6 to drive OUT3 of PWM-XBAR 0: Respective output of MUX6 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX6 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of MUX5 to drive OUT3 of PWM-XBAR 0: Respective output of MUX5 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX5 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of MUX4 to drive OUT3 of PWM-XBAR 0: Respective output of MUX4 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX4 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of MUX3 to drive OUT3 of PWM-XBAR 0: Respective output of MUX3 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX3 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	MUX2	R/W	0h	Selects the output of MUX2 to drive OUT3 of PWM-XBAR 0: Respective output of MUX2 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX2 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of MUX1 to drive OUT3 of PWM-XBAR 0: Respective output of MUX1 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX1 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of MUX0 to drive OUT3 of PWM-XBAR 0: Respective output of MUX0 is disabled to drive the OUT3 of PWM-XBAR 1: Respective output of MUX0 is enabled to drive the OUT3 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.20 TRIP4MUXENABLE Register (Offset = 26h) [Reset = 00000000h]

TRIP4MUXENABLE is shown in Figure 9-51 and described in Table 9-57.

Return to the Summary Table.

PWM XBAR Mux Enable for Output4

Figure 9-51. TRIP4MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	MUX17	MUX16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
MUX15	MUX14	MUX13	MUX12	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-57. TRIP4MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	MUX17	R/W	0h	Selects the output of MUX17 to drive OUT4 of PWM-XBAR 0: Respective output of MUX17 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX17 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
16	MUX16	R/W	0h	Selects the output of MUX16 to drive OUT4 of PWM-XBAR 0: Respective output of MUX16 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX16 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-57. TRIP4MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	MUX15	R/W	0h	Selects the output of MUX15 to drive OUT4 of PWM-XBAR 0: Respective output of MUX15 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX15 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
14	MUX14	R/W	0h	Selects the output of MUX14 to drive OUT4 of PWM-XBAR 0: Respective output of MUX14 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX14 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13	MUX13	R/W	0h	Selects the output of MUX13 to drive OUT4 of PWM-XBAR 0: Respective output of MUX13 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX13 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
12	MUX12	R/W	0h	Selects the output of MUX12 to drive OUT4 of PWM-XBAR 0: Respective output of MUX12 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX12 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11	MUX11	R/W	0h	Selects the output of MUX11 to drive OUT4 of PWM-XBAR 0: Respective output of MUX11 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX11 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
10	MUX10	R/W	0h	Selects the output of MUX10 to drive OUT4 of PWM-XBAR 0: Respective output of MUX10 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX10 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of MUX9 to drive OUT4 of PWM-XBAR 0: Respective output of MUX9 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX9 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of MUX8 to drive OUT4 of PWM-XBAR 0: Respective output of MUX8 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX8 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-57. TRIP4MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	MUX7	R/W	0h	Selects the output of MUX7 to drive OUT4 of PWM-XBAR 0: Respective output of MUX7 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX7 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of MUX6 to drive OUT4 of PWM-XBAR 0: Respective output of MUX6 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX6 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of MUX5 to drive OUT4 of PWM-XBAR 0: Respective output of MUX5 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX5 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of MUX4 to drive OUT4 of PWM-XBAR 0: Respective output of MUX4 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX4 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of MUX3 to drive OUT4 of PWM-XBAR 0: Respective output of MUX3 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX3 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	MUX2	R/W	0h	Selects the output of MUX2 to drive OUT4 of PWM-XBAR 0: Respective output of MUX2 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX2 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of MUX1 to drive OUT4 of PWM-XBAR 0: Respective output of MUX1 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX1 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of MUX0 to drive OUT4 of PWM-XBAR 0: Respective output of MUX0 is disabled to drive the OUT4 of PWM-XBAR 1: Respective output of MUX0 is enabled to drive the OUT4 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.21 TRIP5MUXENABLE Register (Offset = 28h) [Reset = 00000000h]

TRIP5MUXENABLE is shown in Figure 9-52 and described in Table 9-58.

Return to the Summary Table.

PWM XBAR Mux Enable for Output5

Figure 9-52. TRIP5MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	MUX17	MUX16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
MUX15	MUX14	MUX13	MUX12	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-58. TRIP5MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	MUX17	R/W	0h	Selects the output of MUX17 to drive OUT5 of PWM-XBAR 0: Respective output of MUX17 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX17 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
16	MUX16	R/W	0h	Selects the output of MUX16 to drive OUT5 of PWM-XBAR 0: Respective output of MUX16 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX16 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-58. TRIP5MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	MUX15	R/W	0h	Selects the output of MUX15 to drive OUT5 of PWM-XBAR 0: Respective output of MUX15 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX15 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
14	MUX14	R/W	0h	Selects the output of MUX14 to drive OUT5 of PWM-XBAR 0: Respective output of MUX14 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX14 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13	MUX13	R/W	0h	Selects the output of MUX13 to drive OUT5 of PWM-XBAR 0: Respective output of MUX13 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX13 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
12	MUX12	R/W	0h	Selects the output of MUX12 to drive OUT5 of PWM-XBAR 0: Respective output of MUX12 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX12 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11	MUX11	R/W	0h	Selects the output of MUX11 to drive OUT5 of PWM-XBAR 0: Respective output of MUX11 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX11 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
10	MUX10	R/W	0h	Selects the output of MUX10 to drive OUT5 of PWM-XBAR 0: Respective output of MUX10 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX10 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of MUX9 to drive OUT5 of PWM-XBAR 0: Respective output of MUX9 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX9 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of MUX8 to drive OUT5 of PWM-XBAR 0: Respective output of MUX8 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX8 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-58. TRIP5MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	MUX7	R/W	0h	Selects the output of MUX7 to drive OUT5 of PWM-XBAR 0: Respective output of MUX7 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX7 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of MUX6 to drive OUT5 of PWM-XBAR 0: Respective output of MUX6 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX6 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of MUX5 to drive OUT5 of PWM-XBAR 0: Respective output of MUX5 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX5 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of MUX4 to drive OUT5 of PWM-XBAR 0: Respective output of MUX4 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX4 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of MUX3 to drive OUT5 of PWM-XBAR 0: Respective output of MUX3 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX3 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	MUX2	R/W	0h	Selects the output of MUX2 to drive OUT5 of PWM-XBAR 0: Respective output of MUX2 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX2 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of MUX1 to drive OUT5 of PWM-XBAR 0: Respective output of MUX1 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX1 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of MUX0 to drive OUT5 of PWM-XBAR 0: Respective output of MUX0 is disabled to drive the OUT5 of PWM-XBAR 1: Respective output of MUX0 is enabled to drive the OUT5 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.22 TRIP6MUXENABLE Register (Offset = 2Ah) [Reset = 00000000h]

TRIP6MUXENABLE is shown in Figure 9-53 and described in Table 9-59.

Return to the Summary Table.

PWM XBAR Mux Enable for Output6

Figure 9-53. TRIP6MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	MUX17	MUX16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
MUX15	MUX14	MUX13	MUX12	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-59. TRIP6MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	MUX17	R/W	0h	Selects the output of MUX17 to drive OUT6 of PWM-XBAR 0: Respective output of MUX17 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX17 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
16	MUX16	R/W	0h	Selects the output of MUX16 to drive OUT6 of PWM-XBAR 0: Respective output of MUX16 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX16 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-59. TRIP6MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	MUX15	R/W	0h	Selects the output of MUX15 to drive OUT6 of PWM-XBAR 0: Respective output of MUX15 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX15 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
14	MUX14	R/W	0h	Selects the output of MUX14 to drive OUT6 of PWM-XBAR 0: Respective output of MUX14 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX14 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13	MUX13	R/W	0h	Selects the output of MUX13 to drive OUT6 of PWM-XBAR 0: Respective output of MUX13 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX13 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
12	MUX12	R/W	0h	Selects the output of MUX12 to drive OUT6 of PWM-XBAR 0: Respective output of MUX12 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX12 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11	MUX11	R/W	0h	Selects the output of MUX11 to drive OUT6 of PWM-XBAR 0: Respective output of MUX11 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX11 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
10	MUX10	R/W	0h	Selects the output of MUX10 to drive OUT6 of PWM-XBAR 0: Respective output of MUX10 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX10 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of MUX9 to drive OUT6 of PWM-XBAR 0: Respective output of MUX9 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX9 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of MUX8 to drive OUT6 of PWM-XBAR 0: Respective output of MUX8 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX8 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-59. TRIP6MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	MUX7	R/W	0h	Selects the output of MUX7 to drive OUT6 of PWM-XBAR 0: Respective output of MUX7 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX7 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of MUX6 to drive OUT6 of PWM-XBAR 0: Respective output of MUX6 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX6 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of MUX5 to drive OUT6 of PWM-XBAR 0: Respective output of MUX5 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX5 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of MUX4 to drive OUT6 of PWM-XBAR 0: Respective output of MUX4 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX4 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of MUX3 to drive OUT6 of PWM-XBAR 0: Respective output of MUX3 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX3 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	MUX2	R/W	0h	Selects the output of MUX2 to drive OUT6 of PWM-XBAR 0: Respective output of MUX2 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX2 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of MUX1 to drive OUT6 of PWM-XBAR 0: Respective output of MUX1 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX1 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of MUX0 to drive OUT6 of PWM-XBAR 0: Respective output of MUX0 is disabled to drive the OUT6 of PWM-XBAR 1: Respective output of MUX0 is enabled to drive the OUT6 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.23 TRIP7MUXENABLE Register (Offset = 2Ch) [Reset = 00000000h]

TRIP7MUXENABLE is shown in Figure 9-54 and described in Table 9-60.

Return to the Summary Table.

PWM XBAR Mux Enable for Output7

Figure 9-54. TRIP7MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	MUX17	MUX16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
MUX15	MUX14	MUX13	MUX12	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-60. TRIP7MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	MUX17	R/W	0h	Selects the output of MUX17 to drive OUT7 of PWM-XBAR 0: Respective output of MUX17 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX17 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
16	MUX16	R/W	0h	Selects the output of MUX16 to drive OUT7 of PWM-XBAR 0: Respective output of MUX16 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX16 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-60. TRIP7MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	MUX15	R/W	0h	Selects the output of MUX15 to drive OUT7 of PWM-XBAR 0: Respective output of MUX15 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX15 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
14	MUX14	R/W	0h	Selects the output of MUX14 to drive OUT7 of PWM-XBAR 0: Respective output of MUX14 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX14 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13	MUX13	R/W	0h	Selects the output of MUX13 to drive OUT7 of PWM-XBAR 0: Respective output of MUX13 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX13 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
12	MUX12	R/W	0h	Selects the output of MUX12 to drive OUT7 of PWM-XBAR 0: Respective output of MUX12 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX12 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11	MUX11	R/W	0h	Selects the output of MUX11 to drive OUT7 of PWM-XBAR 0: Respective output of MUX11 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX11 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
10	MUX10	R/W	0h	Selects the output of MUX10 to drive OUT7 of PWM-XBAR 0: Respective output of MUX10 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX10 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of MUX9 to drive OUT7 of PWM-XBAR 0: Respective output of MUX9 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX9 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of MUX8 to drive OUT7 of PWM-XBAR 0: Respective output of MUX8 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX8 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-60. TRIP7MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	MUX7	R/W	0h	Selects the output of MUX7 to drive OUT7 of PWM-XBAR 0: Respective output of MUX7 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX7 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of MUX6 to drive OUT7 of PWM-XBAR 0: Respective output of MUX6 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX6 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of MUX5 to drive OUT7 of PWM-XBAR 0: Respective output of MUX5 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX5 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of MUX4 to drive OUT7 of PWM-XBAR 0: Respective output of MUX4 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX4 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of MUX3 to drive OUT7 of PWM-XBAR 0: Respective output of MUX3 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX3 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	MUX2	R/W	0h	Selects the output of MUX2 to drive OUT7 of PWM-XBAR 0: Respective output of MUX2 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX2 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of MUX1 to drive OUT7 of PWM-XBAR 0: Respective output of MUX1 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX1 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of MUX0 to drive OUT7 of PWM-XBAR 0: Respective output of MUX0 is disabled to drive the OUT7 of PWM-XBAR 1: Respective output of MUX0 is enabled to drive the OUT7 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.24 TRIP8MUXENABLE Register (Offset = 2Eh) [Reset = 00000000h]

TRIP8MUXENABLE is shown in Figure 9-55 and described in Table 9-61.

Return to the Summary Table.

PWM XBAR Mux Enable for Output8

Figure 9-55. TRIP8MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	MUX17	MUX16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
MUX15	MUX14	MUX13	MUX12	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-61. TRIP8MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	MUX17	R/W	0h	Selects the output of MUX17 to drive OUT8 of PWM-XBAR 0: Respective output of MUX17 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX17 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
16	MUX16	R/W	0h	Selects the output of MUX16 to drive OUT8 of PWM-XBAR 0: Respective output of MUX16 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX16 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-61. TRIP8MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	MUX15	R/W	0h	Selects the output of MUX15 to drive OUT8 of PWM-XBAR 0: Respective output of MUX15 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX15 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
14	MUX14	R/W	0h	Selects the output of MUX14 to drive OUT8 of PWM-XBAR 0: Respective output of MUX14 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX14 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
13	MUX13	R/W	0h	Selects the output of MUX13 to drive OUT8 of PWM-XBAR 0: Respective output of MUX13 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX13 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
12	MUX12	R/W	0h	Selects the output of MUX12 to drive OUT8 of PWM-XBAR 0: Respective output of MUX12 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX12 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11	MUX11	R/W	0h	Selects the output of MUX11 to drive OUT8 of PWM-XBAR 0: Respective output of MUX11 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX11 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
10	MUX10	R/W	0h	Selects the output of MUX10 to drive OUT8 of PWM-XBAR 0: Respective output of MUX10 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX10 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of MUX9 to drive OUT8 of PWM-XBAR 0: Respective output of MUX9 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX9 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of MUX8 to drive OUT8 of PWM-XBAR 0: Respective output of MUX8 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX8 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-61. TRIP8MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	MUX7	R/W	0h	Selects the output of MUX7 to drive OUT8 of PWM-XBAR 0: Respective output of MUX7 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX7 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of MUX6 to drive OUT8 of PWM-XBAR 0: Respective output of MUX6 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX6 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of MUX5 to drive OUT8 of PWM-XBAR 0: Respective output of MUX5 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX5 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of MUX4 to drive OUT8 of PWM-XBAR 0: Respective output of MUX4 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX4 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of MUX3 to drive OUT8 of PWM-XBAR 0: Respective output of MUX3 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX3 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	MUX2	R/W	0h	Selects the output of MUX2 to drive OUT8 of PWM-XBAR 0: Respective output of MUX2 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX2 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of MUX1 to drive OUT8 of PWM-XBAR 0: Respective output of MUX1 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX1 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of MUX0 to drive OUT8 of PWM-XBAR 0: Respective output of MUX0 is disabled to drive the OUT8 of PWM-XBAR 1: Respective output of MUX0 is enabled to drive the OUT8 of PWM-XBAR Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.25 TRIPOUTINV Register (Offset = 38h) [Reset = 00000000h]

TRIPOUTINV is shown in Figure 9-56 and described in Table 9-62.

Return to the Summary Table.

PWM XBAR Output Inversion Register

Figure 9-56. TRIPOUTINV Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
OUT8	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-62. TRIPOUTINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	RESERVED	R-0	0h	Reserved
7	OUT8	R/W	0h	Selects polarity for OUT8 of PWM-XBAR 0: drives active high output 1: drives active-low output Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	OUT7	R/W	0h	Selects polarity for OUT7 of PWM-XBAR 0: drives active high output 1: drives active-low output Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	OUT6	R/W	0h	Selects polarity for OUT6 of PWM-XBAR 0: drives active high output 1: drives active-low output Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	OUT5	R/W	0h	Selects polarity for OUT5 of PWM-XBAR 0: drives active high output 1: drives active-low output Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	OUT4	R/W	0h	Selects polarity for OUT4 of PWM-XBAR 0: drives active high output 1: drives active-low output Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	OUT3	R/W	0h	Selects polarity for OUT3 of PWM-XBAR 0: drives active high output 1: drives active-low output Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-62. TRIPOUTINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	OUT2	R/W	0h	Selects polarity for OUT2 of PWM-XBAR 0: drives active high output 1: drives active-low output Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	OUT1	R/W	0h	Selects polarity for OUT1 of PWM-XBAR 0: drives active high output 1: drives active-low output Refer to the PWM X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.4.26 TRIPLOCK Register (Offset = 3Eh) [Reset = 00000000h]

TRIPLOCK is shown in Figure 9-57 and described in Table 9-63.

Return to the Summary Table.

PWM XBAR Configuration Lock register

Figure 9-57. TRIPLOCK Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							LOCK
R-0-0h							R/WSonce-0h

Table 9-63. TRIPLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Bit-0 of this register can be set only if KEY= 0x5a5a Reset type: CPU1.SYSRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/WSonce	0h	Locks the configuration for PWM-XBAR. Once the configuration is locked, writes to the below registers for PWM-XBAR is blocked. Registers Affected by the LOCK mechanism: PWM-XBAROUTyMUX0TO15CFG PWM-XBAROUTyMUX16TO31CFG PWM-XBAROUTyMUXENABLE PWM-XBAROUTLATEN PWM-XBAROUTINV 0: Writes to the above registers are allowed 1: Writes to the above registers are blocked Note: [1] LOCK mechanism only apples to writes. Reads are never blocked. Reset type: CPU1.SYSRSn

9.3.5 OUTPUT_XBAR_REGS Registers

Table 9-64 lists the memory-mapped registers for the OUTPUT_XBAR_REGS registers. All register offset addresses not listed in Table 9-64 should be considered as reserved locations and the register contents should not be modified.

Table 9-64. OUTPUT_XBAR_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	OUTPUT1MUX0TO15CFG	Output X-BAR Mux Configuration for Output 1	EALLOW
4h	OUTPUT2MUX0TO15CFG	Output X-BAR Mux Configuration for Output 2	EALLOW
8h	OUTPUT3MUX0TO15CFG	Output X-BAR Mux Configuration for Output 3	EALLOW
Ch	OUTPUT4MUX0TO15CFG	Output X-BAR Mux Configuration for Output 4	EALLOW
10h	OUTPUT5MUX0TO15CFG	Output X-BAR Mux Configuration for Output 5	EALLOW
14h	OUTPUT6MUX0TO15CFG	Output X-BAR Mux Configuration for Output 6	EALLOW
18h	OUTPUT7MUX0TO15CFG	Output X-BAR Mux Configuration for Output 7	EALLOW
1Ch	OUTPUT8MUX0TO15CFG	Output X-BAR Mux Configuration for Output 8	EALLOW
20h	OUTPUT1MUXENABLE	Output X-BAR Mux Enable for Output 1	EALLOW
22h	OUTPUT2MUXENABLE	Output X-BAR Mux Enable for Output 2	EALLOW
24h	OUTPUT3MUXENABLE	Output X-BAR Mux Enable for Output 3	EALLOW
26h	OUTPUT4MUXENABLE	Output X-BAR Mux Enable for Output 4	EALLOW
28h	OUTPUT5MUXENABLE	Output X-BAR Mux Enable for Output 5	EALLOW
2Ah	OUTPUT6MUXENABLE	Output X-BAR Mux Enable for Output 6	EALLOW
2Ch	OUTPUT7MUXENABLE	Output X-BAR Mux Enable for Output 7	EALLOW
2Eh	OUTPUT8MUXENABLE	Output X-BAR Mux Enable for Output 8	EALLOW
30h	OUTPUTLATCH	Output X-BAR Output Latch	
32h	OUTPUTLATCHCLR	Output X-BAR Output Latch Clear	
34h	OUTPUTLATCHFRC	Output X-BAR Output Latch Clear	
36h	OUTPUTLATCHENABLE	Output X-BAR Output Latch Enable	EALLOW
38h	OUTPUTINV	Output X-BAR Output Inversion	EALLOW
3Eh	OUTPUTLOCK	Output X-BAR Configuration Lock register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 9-65 shows the codes that are used for access types in this section.

Table 9-65. OUTPUT_XBAR_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 9-65. OUTPUT_XBAR_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

9.3.5.1 OUTPUT1MUX0TO15CFG Register (Offset = 0h) [Reset = 00000000h]

OUTPUT1MUX0TO15CFG is shown in Figure 9-58 and described in Table 9-66.

Return to the Summary Table.

Output X-BAR Mux Configuration for Output 1

Figure 9-58. OUTPUT1MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-66. OUTPUT1MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	MUX11	R/W	0h	Select Bits for OUTPUT1 Mux11: 00 : Select .0 input for Mux11 01 : Select .1 input for Mux11 10 : Select .2 input for Mux11 11 : Select .3 input for Mux11 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for OUTPUT1 Mux10: 00 : Select .0 input for Mux10 01 : Select .1 input for Mux10 10 : Select .2 input for Mux10 11 : Select .3 input for Mux10 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
19-18	MUX9	R/W	0h	Select Bits for OUTPUT1 Mux9: 00 : Select .0 input for Mux9 01 : Select .1 input for Mux9 10 : Select .2 input for Mux9 11 : Select .3 input for Mux9 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for OUTPUT1 Mux8: 00 : Select .0 input for Mux8 01 : Select .1 input for Mux8 10 : Select .2 input for Mux8 11 : Select .3 input for Mux8 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for OUTPUT1 Mux7: 00 : Select .0 input for Mux7 01 : Select .1 input for Mux7 10 : Select .2 input for Mux7 11 : Select .3 input for Mux7 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-66. OUTPUT1MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	MUX6	R/W	0h	Select Bits for OUTPUT1 Mux6: 00 : Select .0 input for Mux6 01 : Select .1 input for Mux6 10 : Select .2 input for Mux6 11 : Select .3 input for Mux6 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for OUTPUT1 Mux5: 00 : Select .0 input for Mux5 01 : Select .1 input for Mux5 10 : Select .2 input for Mux5 11 : Select .3 input for Mux5 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for OUTPUT1 Mux4: 00 : Select .0 input for Mux4 01 : Select .1 input for Mux4 10 : Select .2 input for Mux4 11 : Select .3 input for Mux4 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for OUTPUT1 Mux3: 00 : Select .0 input for Mux3 01 : Select .1 input for Mux3 10 : Select .2 input for Mux3 11 : Select .3 input for Mux3 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for OUTPUT1 Mux2: 00 : Select .0 input for Mux2 01 : Select .1 input for Mux2 10 : Select .2 input for Mux2 11 : Select .3 input for Mux2 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3-2	MUX1	R/W	0h	Select Bits for OUTPUT1 Mux1: 00 : Select .0 input for Mux1 01 : Select .1 input for Mux1 10 : Select .2 input for Mux1 11 : Select .3 input for Mux1 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for OUTPUT1 Mux0: 00 : Select .0 input for Mux0 01 : Select .1 input for Mux0 10 : Select .2 input for Mux0 11 : Select .3 input for Mux0 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.2 OUTPUT2MUX0TO15CFG Register (Offset = 4h) [Reset = 00000000h]

OUTPUT2MUX0TO15CFG is shown in Figure 9-59 and described in Table 9-67.

Return to the Summary Table.

Output X-BAR Mux Configuration for Output 2

Figure 9-59. OUTPUT2MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-67. OUTPUT2MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	MUX11	R/W	0h	Select Bits for OUTPUT2 Mux11: 00 : Select .0 input for Mux11 01 : Select .1 input for Mux11 10 : Select .2 input for Mux11 11 : Select .3 input for Mux11 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for OUTPUT2 Mux10: 00 : Select .0 input for Mux10 01 : Select .1 input for Mux10 10 : Select .2 input for Mux10 11 : Select .3 input for Mux10 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
19-18	MUX9	R/W	0h	Select Bits for OUTPUT2 Mux9: 00 : Select .0 input for Mux9 01 : Select .1 input for Mux9 10 : Select .2 input for Mux9 11 : Select .3 input for Mux9 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for OUTPUT2 Mux8: 00 : Select .0 input for Mux8 01 : Select .1 input for Mux8 10 : Select .2 input for Mux8 11 : Select .3 input for Mux8 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for OUTPUT2 Mux7: 00 : Select .0 input for Mux7 01 : Select .1 input for Mux7 10 : Select .2 input for Mux7 11 : Select .3 input for Mux7 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-67. OUTPUT2MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	MUX6	R/W	0h	Select Bits for OUTPUT2 Mux6: 00 : Select .0 input for Mux6 01 : Select .1 input for Mux6 10 : Select .2 input for Mux6 11 : Select .3 input for Mux6 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for OUTPUT2 Mux5: 00 : Select .0 input for Mux5 01 : Select .1 input for Mux5 10 : Select .2 input for Mux5 11 : Select .3 input for Mux5 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for OUTPUT2 Mux4: 00 : Select .0 input for Mux4 01 : Select .1 input for Mux4 10 : Select .2 input for Mux4 11 : Select .3 input for Mux4 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for OUTPUT2 Mux3: 00 : Select .0 input for Mux3 01 : Select .1 input for Mux3 10 : Select .2 input for Mux3 11 : Select .3 input for Mux3 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for OUTPUT2 Mux2: 00 : Select .0 input for Mux2 01 : Select .1 input for Mux2 10 : Select .2 input for Mux2 11 : Select .3 input for Mux2 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3-2	MUX1	R/W	0h	Select Bits for OUTPUT2 Mux1: 00 : Select .0 input for Mux1 01 : Select .1 input for Mux1 10 : Select .2 input for Mux1 11 : Select .3 input for Mux1 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for OUTPUT2 Mux0: 00 : Select .0 input for Mux0 01 : Select .1 input for Mux0 10 : Select .2 input for Mux0 11 : Select .3 input for Mux0 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.3 OUTPUT3MUX0TO15CFG Register (Offset = 8h) [Reset = 00000000h]

OUTPUT3MUX0TO15CFG is shown in Figure 9-60 and described in Table 9-68.

Return to the Summary Table.

Output X-BAR Mux Configuration for Output 3

Figure 9-60. OUTPUT3MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-68. OUTPUT3MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	MUX11	R/W	0h	Select Bits for OUTPUT3 Mux11: 00 : Select .0 input for Mux11 01 : Select .1 input for Mux11 10 : Select .2 input for Mux11 11 : Select .3 input for Mux11 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for OUTPUT3 Mux10: 00 : Select .0 input for Mux10 01 : Select .1 input for Mux10 10 : Select .2 input for Mux10 11 : Select .3 input for Mux10 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
19-18	MUX9	R/W	0h	Select Bits for OUTPUT3 Mux9: 00 : Select .0 input for Mux9 01 : Select .1 input for Mux9 10 : Select .2 input for Mux9 11 : Select .3 input for Mux9 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for OUTPUT3 Mux8: 00 : Select .0 input for Mux8 01 : Select .1 input for Mux8 10 : Select .2 input for Mux8 11 : Select .3 input for Mux8 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for OUTPUT3 Mux7: 00 : Select .0 input for Mux7 01 : Select .1 input for Mux7 10 : Select .2 input for Mux7 11 : Select .3 input for Mux7 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-68. OUTPUT3MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	MUX6	R/W	0h	Select Bits for OUTPUT3 Mux6: 00 : Select .0 input for Mux6 01 : Select .1 input for Mux6 10 : Select .2 input for Mux6 11 : Select .3 input for Mux6 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for OUTPUT3 Mux5: 00 : Select .0 input for Mux5 01 : Select .1 input for Mux5 10 : Select .2 input for Mux5 11 : Select .3 input for Mux5 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for OUTPUT3 Mux4: 00 : Select .0 input for Mux4 01 : Select .1 input for Mux4 10 : Select .2 input for Mux4 11 : Select .3 input for Mux4 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for OUTPUT3 Mux3: 00 : Select .0 input for Mux3 01 : Select .1 input for Mux3 10 : Select .2 input for Mux3 11 : Select .3 input for Mux3 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for OUTPUT3 Mux2: 00 : Select .0 input for Mux2 01 : Select .1 input for Mux2 10 : Select .2 input for Mux2 11 : Select .3 input for Mux2 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3-2	MUX1	R/W	0h	Select Bits for OUTPUT3 Mux1: 00 : Select .0 input for Mux1 01 : Select .1 input for Mux1 10 : Select .2 input for Mux1 11 : Select .3 input for Mux1 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for OUTPUT3 Mux0: 00 : Select .0 input for Mux0 01 : Select .1 input for Mux0 10 : Select .2 input for Mux0 11 : Select .3 input for Mux0 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.4 OUTPUT4MUX0TO15CFG Register (Offset = Ch) [Reset = 00000000h]

OUTPUT4MUX0TO15CFG is shown in Figure 9-61 and described in Table 9-69.

Return to the Summary Table.

Output X-BAR Mux Configuration for Output 4

Figure 9-61. OUTPUT4MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-69. OUTPUT4MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	MUX11	R/W	0h	Select Bits for OUTPUT4 Mux11: 00 : Select .0 input for Mux11 01 : Select .1 input for Mux11 10 : Select .2 input for Mux11 11 : Select .3 input for Mux11 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for OUTPUT4 Mux10: 00 : Select .0 input for Mux10 01 : Select .1 input for Mux10 10 : Select .2 input for Mux10 11 : Select .3 input for Mux10 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
19-18	MUX9	R/W	0h	Select Bits for OUTPUT4 Mux9: 00 : Select .0 input for Mux9 01 : Select .1 input for Mux9 10 : Select .2 input for Mux9 11 : Select .3 input for Mux9 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for OUTPUT4 Mux8: 00 : Select .0 input for Mux8 01 : Select .1 input for Mux8 10 : Select .2 input for Mux8 11 : Select .3 input for Mux8 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for OUTPUT4 Mux7: 00 : Select .0 input for Mux7 01 : Select .1 input for Mux7 10 : Select .2 input for Mux7 11 : Select .3 input for Mux7 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-69. OUTPUT4MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	MUX6	R/W	0h	Select Bits for OUTPUT4 Mux6: 00 : Select .0 input for Mux6 01 : Select .1 input for Mux6 10 : Select .2 input for Mux6 11 : Select .3 input for Mux6 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for OUTPUT4 Mux5: 00 : Select .0 input for Mux5 01 : Select .1 input for Mux5 10 : Select .2 input for Mux5 11 : Select .3 input for Mux5 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for OUTPUT4 Mux4: 00 : Select .0 input for Mux4 01 : Select .1 input for Mux4 10 : Select .2 input for Mux4 11 : Select .3 input for Mux4 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for OUTPUT4 Mux3: 00 : Select .0 input for Mux3 01 : Select .1 input for Mux3 10 : Select .2 input for Mux3 11 : Select .3 input for Mux3 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for OUTPUT4 Mux2: 00 : Select .0 input for Mux2 01 : Select .1 input for Mux2 10 : Select .2 input for Mux2 11 : Select .3 input for Mux2 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3-2	MUX1	R/W	0h	Select Bits for OUTPUT4 Mux1: 00 : Select .0 input for Mux1 01 : Select .1 input for Mux1 10 : Select .2 input for Mux1 11 : Select .3 input for Mux1 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for OUTPUT4 Mux0: 00 : Select .0 input for Mux0 01 : Select .1 input for Mux0 10 : Select .2 input for Mux0 11 : Select .3 input for Mux0 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.5 OUTPUT5MUX0TO15CFG Register (Offset = 10h) [Reset = 00000000h]

OUTPUT5MUX0TO15CFG is shown in Figure 9-62 and described in Table 9-70.

Return to the Summary Table.

Output X-BAR Mux Configuration for Output 5

Figure 9-62. OUTPUT5MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-70. OUTPUT5MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	MUX11	R/W	0h	Select Bits for OUTPUT5 Mux11: 00 : Select .0 input for Mux11 01 : Select .1 input for Mux11 10 : Select .2 input for Mux11 11 : Select .3 input for Mux11 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for OUTPUT5 Mux10: 00 : Select .0 input for Mux10 01 : Select .1 input for Mux10 10 : Select .2 input for Mux10 11 : Select .3 input for Mux10 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
19-18	MUX9	R/W	0h	Select Bits for OUTPUT5 Mux9: 00 : Select .0 input for Mux9 01 : Select .1 input for Mux9 10 : Select .2 input for Mux9 11 : Select .3 input for Mux9 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for OUTPUT5 Mux8: 00 : Select .0 input for Mux8 01 : Select .1 input for Mux8 10 : Select .2 input for Mux8 11 : Select .3 input for Mux8 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for OUTPUT5 Mux7: 00 : Select .0 input for Mux7 01 : Select .1 input for Mux7 10 : Select .2 input for Mux7 11 : Select .3 input for Mux7 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-70. OUTPUT5MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	MUX6	R/W	0h	Select Bits for OUTPUT5 Mux6: 00 : Select .0 input for Mux6 01 : Select .1 input for Mux6 10 : Select .2 input for Mux6 11 : Select .3 input for Mux6 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for OUTPUT5 Mux5: 00 : Select .0 input for Mux5 01 : Select .1 input for Mux5 10 : Select .2 input for Mux5 11 : Select .3 input for Mux5 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for OUTPUT5 Mux4: 00 : Select .0 input for Mux4 01 : Select .1 input for Mux4 10 : Select .2 input for Mux4 11 : Select .3 input for Mux4 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for OUTPUT5 Mux3: 00 : Select .0 input for Mux3 01 : Select .1 input for Mux3 10 : Select .2 input for Mux3 11 : Select .3 input for Mux3 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for OUTPUT5 Mux2: 00 : Select .0 input for Mux2 01 : Select .1 input for Mux2 10 : Select .2 input for Mux2 11 : Select .3 input for Mux2 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3-2	MUX1	R/W	0h	Select Bits for OUTPUT5 Mux1: 00 : Select .0 input for Mux1 01 : Select .1 input for Mux1 10 : Select .2 input for Mux1 11 : Select .3 input for Mux1 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for OUTPUT5 Mux0: 00 : Select .0 input for Mux0 01 : Select .1 input for Mux0 10 : Select .2 input for Mux0 11 : Select .3 input for Mux0 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.6 OUTPUT6MUX0TO15CFG Register (Offset = 14h) [Reset = 00000000h]

OUTPUT6MUX0TO15CFG is shown in Figure 9-63 and described in Table 9-71.

Return to the Summary Table.

Output X-BAR Mux Configuration for Output 6

Figure 9-63. OUTPUT6MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-71. OUTPUT6MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	MUX11	R/W	0h	Select Bits for OUTPUT6 Mux11: 00 : Select .0 input for Mux11 01 : Select .1 input for Mux11 10 : Select .2 input for Mux11 11 : Select .3 input for Mux11 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for OUTPUT6 Mux10: 00 : Select .0 input for Mux10 01 : Select .1 input for Mux10 10 : Select .2 input for Mux10 11 : Select .3 input for Mux10 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
19-18	MUX9	R/W	0h	Select Bits for OUTPUT6 Mux9: 00 : Select .0 input for Mux9 01 : Select .1 input for Mux9 10 : Select .2 input for Mux9 11 : Select .3 input for Mux9 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for OUTPUT6 Mux8: 00 : Select .0 input for Mux8 01 : Select .1 input for Mux8 10 : Select .2 input for Mux8 11 : Select .3 input for Mux8 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for OUTPUT6 Mux7: 00 : Select .0 input for Mux7 01 : Select .1 input for Mux7 10 : Select .2 input for Mux7 11 : Select .3 input for Mux7 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-71. OUTPUT6MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	MUX6	R/W	0h	Select Bits for OUTPUT6 Mux6: 00 : Select .0 input for Mux6 01 : Select .1 input for Mux6 10 : Select .2 input for Mux6 11 : Select .3 input for Mux6 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for OUTPUT6 Mux5: 00 : Select .0 input for Mux5 01 : Select .1 input for Mux5 10 : Select .2 input for Mux5 11 : Select .3 input for Mux5 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for OUTPUT6 Mux4: 00 : Select .0 input for Mux4 01 : Select .1 input for Mux4 10 : Select .2 input for Mux4 11 : Select .3 input for Mux4 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for OUTPUT6 Mux3: 00 : Select .0 input for Mux3 01 : Select .1 input for Mux3 10 : Select .2 input for Mux3 11 : Select .3 input for Mux3 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for OUTPUT6 Mux2: 00 : Select .0 input for Mux2 01 : Select .1 input for Mux2 10 : Select .2 input for Mux2 11 : Select .3 input for Mux2 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3-2	MUX1	R/W	0h	Select Bits for OUTPUT6 Mux1: 00 : Select .0 input for Mux1 01 : Select .1 input for Mux1 10 : Select .2 input for Mux1 11 : Select .3 input for Mux1 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for OUTPUT6 Mux0: 00 : Select .0 input for Mux0 01 : Select .1 input for Mux0 10 : Select .2 input for Mux0 11 : Select .3 input for Mux0 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.7 OUTPUT7MUX0TO15CFG Register (Offset = 18h) [Reset = 00000000h]

OUTPUT7MUX0TO15CFG is shown in Figure 9-64 and described in Table 9-72.

Return to the Summary Table.

Output X-BAR Mux Configuration for Output 7

Figure 9-64. OUTPUT7MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-72. OUTPUT7MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	MUX11	R/W	0h	Select Bits for OUTPUT7 Mux11: 00 : Select .0 input for Mux11 01 : Select .1 input for Mux11 10 : Select .2 input for Mux11 11 : Select .3 input for Mux11 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for OUTPUT7 Mux10: 00 : Select .0 input for Mux10 01 : Select .1 input for Mux10 10 : Select .2 input for Mux10 11 : Select .3 input for Mux10 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
19-18	MUX9	R/W	0h	Select Bits for OUTPUT7 Mux9: 00 : Select .0 input for Mux9 01 : Select .1 input for Mux9 10 : Select .2 input for Mux9 11 : Select .3 input for Mux9 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for OUTPUT7 Mux8: 00 : Select .0 input for Mux8 01 : Select .1 input for Mux8 10 : Select .2 input for Mux8 11 : Select .3 input for Mux8 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for OUTPUT7 Mux7: 00 : Select .0 input for Mux7 01 : Select .1 input for Mux7 10 : Select .2 input for Mux7 11 : Select .3 input for Mux7 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-72. OUTPUT7MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	MUX6	R/W	0h	Select Bits for OUTPUT7 Mux6: 00 : Select .0 input for Mux6 01 : Select .1 input for Mux6 10 : Select .2 input for Mux6 11 : Select .3 input for Mux6 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for OUTPUT7 Mux5: 00 : Select .0 input for Mux5 01 : Select .1 input for Mux5 10 : Select .2 input for Mux5 11 : Select .3 input for Mux5 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for OUTPUT7 Mux4: 00 : Select .0 input for Mux4 01 : Select .1 input for Mux4 10 : Select .2 input for Mux4 11 : Select .3 input for Mux4 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for OUTPUT7 Mux3: 00 : Select .0 input for Mux3 01 : Select .1 input for Mux3 10 : Select .2 input for Mux3 11 : Select .3 input for Mux3 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for OUTPUT7 Mux2: 00 : Select .0 input for Mux2 01 : Select .1 input for Mux2 10 : Select .2 input for Mux2 11 : Select .3 input for Mux2 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3-2	MUX1	R/W	0h	Select Bits for OUTPUT7 Mux1: 00 : Select .0 input for Mux1 01 : Select .1 input for Mux1 10 : Select .2 input for Mux1 11 : Select .3 input for Mux1 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for OUTPUT7 Mux0: 00 : Select .0 input for Mux0 01 : Select .1 input for Mux0 10 : Select .2 input for Mux0 11 : Select .3 input for Mux0 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.8 OUTPUT8MUX0TO15CFG Register (Offset = 1Ch) [Reset = 00000000h]

OUTPUT8MUX0TO15CFG is shown in Figure 9-65 and described in Table 9-73.

Return to the Summary Table.

Output X-BAR Mux Configuration for Output 8

Figure 9-65. OUTPUT8MUX0TO15CFG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED		RESERVED		RESERVED		RESERVED		MUX11		MUX10		MUX9		MUX8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX7		MUX6		MUX5		MUX4		MUX3		MUX2		MUX1		MUX0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 9-73. OUTPUT8MUX0TO15CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	MUX11	R/W	0h	Select Bits for OUTPUT8 Mux11: 00 : Select .0 input for Mux11 01 : Select .1 input for Mux11 10 : Select .2 input for Mux11 11 : Select .3 input for Mux11 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
21-20	MUX10	R/W	0h	Select Bits for OUTPUT8 Mux10: 00 : Select .0 input for Mux10 01 : Select .1 input for Mux10 10 : Select .2 input for Mux10 11 : Select .3 input for Mux10 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
19-18	MUX9	R/W	0h	Select Bits for OUTPUT8 Mux9: 00 : Select .0 input for Mux9 01 : Select .1 input for Mux9 10 : Select .2 input for Mux9 11 : Select .3 input for Mux9 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
17-16	MUX8	R/W	0h	Select Bits for OUTPUT8 Mux8: 00 : Select .0 input for Mux8 01 : Select .1 input for Mux8 10 : Select .2 input for Mux8 11 : Select .3 input for Mux8 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
15-14	MUX7	R/W	0h	Select Bits for OUTPUT8 Mux7: 00 : Select .0 input for Mux7 01 : Select .1 input for Mux7 10 : Select .2 input for Mux7 11 : Select .3 input for Mux7 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-73. OUTPUT8MUX0TO15CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	MUX6	R/W	0h	Select Bits for OUTPUT8 Mux6: 00 : Select .0 input for Mux6 01 : Select .1 input for Mux6 10 : Select .2 input for Mux6 11 : Select .3 input for Mux6 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
11-10	MUX5	R/W	0h	Select Bits for OUTPUT8 Mux5: 00 : Select .0 input for Mux5 01 : Select .1 input for Mux5 10 : Select .2 input for Mux5 11 : Select .3 input for Mux5 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9-8	MUX4	R/W	0h	Select Bits for OUTPUT8 Mux4: 00 : Select .0 input for Mux4 01 : Select .1 input for Mux4 10 : Select .2 input for Mux4 11 : Select .3 input for Mux4 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7-6	MUX3	R/W	0h	Select Bits for OUTPUT8 Mux3: 00 : Select .0 input for Mux3 01 : Select .1 input for Mux3 10 : Select .2 input for Mux3 11 : Select .3 input for Mux3 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5-4	MUX2	R/W	0h	Select Bits for OUTPUT8 Mux2: 00 : Select .0 input for Mux2 01 : Select .1 input for Mux2 10 : Select .2 input for Mux2 11 : Select .3 input for Mux2 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3-2	MUX1	R/W	0h	Select Bits for OUTPUT8 Mux1: 00 : Select .0 input for Mux1 01 : Select .1 input for Mux1 10 : Select .2 input for Mux1 11 : Select .3 input for Mux1 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1-0	MUX0	R/W	0h	Select Bits for OUTPUT8 Mux0: 00 : Select .0 input for Mux0 01 : Select .1 input for Mux0 10 : Select .2 input for Mux0 11 : Select .3 input for Mux0 Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.9 OUTPUT1MUXENABLE Register (Offset = 20h) [Reset = 00000000h]

OUTPUT1MUXENABLE is shown in Figure 9-66 and described in Table 9-74.

Return to the Summary Table.

Output X-BAR Mux Enable for Output 1

Figure 9-66. OUTPUT1MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-74. OUTPUT1MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	MUX11	R/W	0h	Selects the output of Mux11 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux11 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux11 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-74. OUTPUT1MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MUX10	R/W	0h	Selects the output of Mux10 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux10 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux10 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of Mux9 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux9 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux9 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of Mux8 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux8 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux8 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7	MUX7	R/W	0h	Selects the output of Mux7 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux7 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux7 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of Mux6 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux6 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux6 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of Mux5 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux5 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux5 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of Mux4 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux4 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux4 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of Mux3 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux3 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux3 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-74. OUTPUT1MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	MUX2	R/W	0h	Selects the output of Mux2 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux2 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux2 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of Mux1 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux1 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux1 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of mux0 to drive OUTPUT1 of OUTPUT-XBAR 0: Respective output of Mux0 is disabled to drive the OUTPUT1 of OUTPUT-XBAR 1: Respective output of Mux0 is enabled to drive the OUTPUT1 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.10 OUTPUT2MUXENABLE Register (Offset = 22h) [Reset = 00000000h]

OUTPUT2MUXENABLE is shown in Figure 9-67 and described in Table 9-75.

Return to the Summary Table.

Output X-BAR Mux Enable for Output 2

Figure 9-67. OUTPUT2MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-75. OUTPUT2MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	MUX11	R/W	0h	Selects the output of Mux11 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux11 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux11 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-75. OUTPUT2MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MUX10	R/W	0h	Selects the output of Mux10 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux10 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux10 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of Mux9 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux9 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux9 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of Mux8 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux8 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux8 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7	MUX7	R/W	0h	Selects the output of Mux7 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux7 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux7 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of Mux6 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux6 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux6 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of Mux5 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux5 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux5 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of Mux4 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux4 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux4 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of Mux3 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux3 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux3 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-75. OUTPUT2MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	MUX2	R/W	0h	Selects the output of Mux2 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux2 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux2 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of Mux1 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux1 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux1 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of mux0 to drive OUTPUT2 of OUTPUT-XBAR 0: Respective output of Mux0 is disabled to drive the OUTPUT2 of OUTPUT-XBAR 1: Respective output of Mux0 is enabled to drive the OUTPUT2 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.11 OUTPUT3MUXENABLE Register (Offset = 24h) [Reset = 00000000h]

OUTPUT3MUXENABLE is shown in Figure 9-68 and described in Table 9-76.

Return to the Summary Table.

Output X-BAR Mux Enable for Output 3

Figure 9-68. OUTPUT3MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-76. OUTPUT3MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	MUX11	R/W	0h	Selects the output of Mux11 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux11 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux11 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-76. OUTPUT3MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MUX10	R/W	0h	Selects the output of Mux10 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux10 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux10 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of Mux9 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux9 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux9 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of Mux8 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux8 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux8 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7	MUX7	R/W	0h	Selects the output of Mux7 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux7 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux7 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of Mux6 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux6 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux6 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of Mux5 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux5 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux5 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of Mux4 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux4 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux4 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of Mux3 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux3 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux3 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-76. OUTPUT3MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	MUX2	R/W	0h	Selects the output of Mux2 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux2 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux2 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of Mux1 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux1 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux1 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of mux0 to drive OUTPUT3 of OUTPUT-XBAR 0: Respective output of Mux0 is disabled to drive the OUTPUT3 of OUTPUT-XBAR 1: Respective output of Mux0 is enabled to drive the OUTPUT3 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.12 OUTPUT4MUXENABLE Register (Offset = 26h) [Reset = 00000000h]

OUTPUT4MUXENABLE is shown in Figure 9-69 and described in Table 9-77.

Return to the Summary Table.

Output X-BAR Mux Enable for Output 4

Figure 9-69. OUTPUT4MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-77. OUTPUT4MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	MUX11	R/W	0h	Selects the output of Mux11 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux11 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux11 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-77. OUTPUT4MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MUX10	R/W	0h	Selects the output of Mux10 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux10 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux10 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of Mux9 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux9 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux9 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of Mux8 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux8 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux8 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7	MUX7	R/W	0h	Selects the output of Mux7 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux7 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux7 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of Mux6 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux6 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux6 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of Mux5 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux5 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux5 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of Mux4 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux4 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux4 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of Mux3 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux3 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux3 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-77. OUTPUT4MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	MUX2	R/W	0h	Selects the output of Mux2 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux2 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux2 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of Mux1 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux1 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux1 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of mux0 to drive OUTPUT4 of OUTPUT-XBAR 0: Respective output of Mux0 is disabled to drive the OUTPUT4 of OUTPUT-XBAR 1: Respective output of Mux0 is enabled to drive the OUTPUT4 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.13 OUTPUT5MUXENABLE Register (Offset = 28h) [Reset = 00000000h]

OUTPUT5MUXENABLE is shown in Figure 9-70 and described in Table 9-78.

Return to the Summary Table.

Output X-BAR Mux Enable for Output 5

Figure 9-70. OUTPUT5MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-78. OUTPUT5MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	MUX11	R/W	0h	Selects the output of Mux11 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux11 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux11 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-78. OUTPUT5MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MUX10	R/W	0h	Selects the output of Mux10 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux10 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux10 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of Mux9 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux9 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux9 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of Mux8 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux8 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux8 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7	MUX7	R/W	0h	Selects the output of Mux7 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux7 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux7 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of Mux6 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux6 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux6 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of Mux5 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux5 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux5 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of Mux4 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux4 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux4 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of Mux3 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux3 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux3 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-78. OUTPUT5MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	MUX2	R/W	0h	Selects the output of Mux2 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux2 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux2 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of Mux1 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux1 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux1 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of mux0 to drive OUTPUT5 of OUTPUT-XBAR 0: Respective output of Mux0 is disabled to drive the OUTPUT5 of OUTPUT-XBAR 1: Respective output of Mux0 is enabled to drive the OUTPUT5 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.14 OUTPUT6MUXENABLE Register (Offset = 2Ah) [Reset = 00000000h]

OUTPUT6MUXENABLE is shown in Figure 9-71 and described in Table 9-79.

Return to the Summary Table.

Output X-BAR Mux Enable for Output 6

Figure 9-71. OUTPUT6MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-79. OUTPUT6MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	MUX11	R/W	0h	Selects the output of Mux11 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux11 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux11 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-79. OUTPUT6MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MUX10	R/W	0h	Selects the output of Mux10 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux10 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux10 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of Mux9 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux9 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux9 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of Mux8 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux8 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux8 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7	MUX7	R/W	0h	Selects the output of Mux7 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux7 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux7 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of Mux6 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux6 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux6 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of Mux5 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux5 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux5 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of Mux4 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux4 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux4 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of Mux3 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux3 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux3 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-79. OUTPUT6MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	MUX2	R/W	0h	Selects the output of Mux2 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux2 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux2 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of Mux1 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux1 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux1 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of mux0 to drive OUTPUT6 of OUTPUT-XBAR 0: Respective output of Mux0 is disabled to drive the OUTPUT6 of OUTPUT-XBAR 1: Respective output of Mux0 is enabled to drive the OUTPUT6 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.15 OUTPUT7MUXENABLE Register (Offset = 2Ch) [Reset = 00000000h]

OUTPUT7MUXENABLE is shown in Figure 9-72 and described in Table 9-80.

Return to the Summary Table.

Output X-BAR Mux Enable for Output 7

Figure 9-72. OUTPUT7MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-80. OUTPUT7MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	MUX11	R/W	0h	Selects the output of Mux11 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux11 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux11 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-80. OUTPUT7MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MUX10	R/W	0h	Selects the output of Mux10 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux10 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux10 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of Mux9 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux9 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux9 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of Mux8 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux8 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux8 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7	MUX7	R/W	0h	Selects the output of Mux7 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux7 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux7 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of Mux6 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux6 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux6 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of Mux5 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux5 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux5 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of Mux4 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux4 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux4 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of Mux3 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux3 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux3 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-80. OUTPUT7MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	MUX2	R/W	0h	Selects the output of Mux2 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux2 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux2 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of Mux1 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux1 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux1 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of mux0 to drive OUTPUT7 of OUTPUT-XBAR 0: Respective output of Mux0 is disabled to drive the OUTPUT7 of OUTPUT-XBAR 1: Respective output of Mux0 is enabled to drive the OUTPUT7 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.16 OUTPUT8MUXENABLE Register (Offset = 2Eh) [Reset = 00000000h]

OUTPUT8MUXENABLE is shown in Figure 9-73 and described in Table 9-81.

Return to the Summary Table.

Output X-BAR Mux Enable for Output 8

Figure 9-73. OUTPUT8MUXENABLE Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	MUX11	MUX10	MUX9	MUX8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
MUX7	MUX6	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-81. OUTPUT8MUXENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	MUX11	R/W	0h	Selects the output of Mux11 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux11 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux11 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-81. OUTPUT8MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MUX10	R/W	0h	Selects the output of Mux10 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux10 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux10 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
9	MUX9	R/W	0h	Selects the output of Mux9 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux9 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux9 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
8	MUX8	R/W	0h	Selects the output of Mux8 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux8 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux8 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
7	MUX7	R/W	0h	Selects the output of Mux7 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux7 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux7 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	MUX6	R/W	0h	Selects the output of Mux6 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux6 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux6 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	MUX5	R/W	0h	Selects the output of Mux5 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux5 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux5 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	MUX4	R/W	0h	Selects the output of Mux4 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux4 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux4 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	MUX3	R/W	0h	Selects the output of Mux3 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux3 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux3 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-81. OUTPUT8MUXENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	MUX2	R/W	0h	Selects the output of Mux2 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux2 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux2 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	MUX1	R/W	0h	Selects the output of Mux1 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux1 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux1 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	MUX0	R/W	0h	Selects the output of mux0 to drive OUTPUT8 of OUTPUT-XBAR 0: Respective output of Mux0 is disabled to drive the OUTPUT8 of OUTPUT-XBAR 1: Respective output of Mux0 is enabled to drive the OUTPUT8 of OUTPUT-XBAR Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.17 OUTPUTLATCH Register (Offset = 30h) [Reset = 00000000h]

OUTPUTLATCH is shown in Figure 9-74 and described in Table 9-82.

Return to the Summary Table.

Output X-BAR Output Latch

Figure 9-74. OUTPUTLATCH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
OUTPUT8	OUTPUT7	OUTPUT6	OUTPUT5	OUTPUT4	OUTPUT3	OUTPUT2	OUTPUT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 9-82. OUTPUTLATCH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	RESERVED	R-0	0h	Reserved
7	OUTPUT8	R	0h	Records the OUTPUT8 of OUTPUT-XBAR. 0: Respective output has not been triggered 1: Respective output is triggered Refer to the Output X-BAR section of this chapter for more details. Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	OUTPUT7	R	0h	Records the OUTPUT7 of OUTPUT-XBAR. 0: Respective output has not been triggered 1: Respective output is triggered Refer to the Output X-BAR section of this chapter for more details. Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	OUTPUT6	R	0h	Records the OUTPUT6 of OUTPUT-XBAR. 0: Respective output has not been triggered 1: Respective output is triggered Refer to the Output X-BAR section of this chapter for more details. Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	OUTPUT5	R	0h	Records the OUTPUT5 of OUTPUT-XBAR. 0: Respective output has not been triggered 1: Respective output is triggered Refer to the Output X-BAR section of this chapter for more details. Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 9-82. OUTPUTLATCH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	OUTPUT4	R	0h	Records the OUTPUT4 of OUTPUT-XBAR. 0: Respective output has not been triggered 1: Respective output is triggered Refer to the Output X-BAR section of this chapter for more details. Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	OUTPUT3	R	0h	Records the OUTPUT3 of OUTPUT-XBAR. 0: Respective output has not been triggered 1: Respective output is triggered Refer to the Output X-BAR section of this chapter for more details. Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	OUTPUT2	R	0h	Records the OUTPUT2 of OUTPUT-XBAR. 0: Respective output has not been triggered 1: Respective output is triggered Refer to the Output X-BAR section of this chapter for more details. Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	OUTPUT1	R	0h	Records the OUTPUT1 of OUTPUT-XBAR. 0: Respective output has not been triggered 1: Respective output is triggered Refer to the Output X-BAR section of this chapter for more details. Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

9.3.5.18 OUTPUTLATCHCLR Register (Offset = 32h) [Reset = 00000000h]

OUTPUTLATCHCLR is shown in Figure 9-75 and described in Table 9-83.

Return to the Summary Table.

Output X-BAR Output Latch Clear

Figure 9-75. OUTPUTLATCHCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
OUTPUT8	OUTPUT7	OUTPUT6	OUTPUT5	OUTPUT4	OUTPUT3	OUTPUT2	OUTPUT1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 9-83. OUTPUTLATCHCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	RESERVED	R-0	0h	Reserved
7	OUTPUT8	R-0/W1S	0h	Clears the Output-Latch for OUTPUT8 of OUTPUT-XBAR Writing 1 clears the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	OUTPUT7	R-0/W1S	0h	Clears the Output-Latch for OUTPUT7 of OUTPUT-XBAR Writing 1 clears the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	OUTPUT6	R-0/W1S	0h	Clears the Output-Latch for OUTPUT6 of OUTPUT-XBAR Writing 1 clears the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	OUTPUT5	R-0/W1S	0h	Clears the Output-Latch for OUTPUT5 of OUTPUT-XBAR Writing 1 clears the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	OUTPUT4	R-0/W1S	0h	Clears the Output-Latch for OUTPUT4 of OUTPUT-XBAR Writing 1 clears the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-83. OUTPUTLATCHCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	OUTPUT3	R-0/W1S	0h	Clears the Output-Latch for OUTPUT3 of OUTPUT-XBAR Writing 1 clears the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	OUTPUT2	R-0/W1S	0h	Clears the Output-Latch for OUTPUT2 of OUTPUT-XBAR Writing 1 clears the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	OUTPUT1	R-0/W1S	0h	Clears the Output-Latch for OUTPUT1 of OUTPUT-XBAR Writing 1 clears the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.19 OUTPUTLATCHFRC Register (Offset = 34h) [Reset = 00000000h]

OUTPUTLATCHFRC is shown in Figure 9-76 and described in Table 9-84.

Return to the Summary Table.

Output X-BAR Output Latch Clear

Figure 9-76. OUTPUTLATCHFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
OUTPUT8	OUTPUT7	OUTPUT6	OUTPUT5	OUTPUT4	OUTPUT3	OUTPUT2	OUTPUT1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 9-84. OUTPUTLATCHFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	RESERVED	R-0	0h	Reserved
7	OUTPUT8	R-0/W1S	0h	Sets the Output-Latch for OUTPUT8 of OUTPUT-XBAR Writing 1 sets the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	OUTPUT7	R-0/W1S	0h	Sets the Output-Latch for OUTPUT7 of OUTPUT-XBAR Writing 1 sets the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	OUTPUT6	R-0/W1S	0h	Sets the Output-Latch for OUTPUT6 of OUTPUT-XBAR Writing 1 sets the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	OUTPUT5	R-0/W1S	0h	Sets the Output-Latch for OUTPUT5 of OUTPUT-XBAR Writing 1 sets the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	OUTPUT4	R-0/W1S	0h	Sets the Output-Latch for OUTPUT4 of OUTPUT-XBAR Writing 1 sets the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-84. OUTPUTLATCHFRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	OUTPUT3	R-0/W1S	0h	Sets the Output-Latch for OUTPUT3 of OUTPUT-XBAR Writing 1 sets the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
1	OUTPUT2	R-0/W1S	0h	Sets the Output-Latch for OUTPUT2 of OUTPUT-XBAR Writing 1 sets the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	OUTPUT1	R-0/W1S	0h	Sets the Output-Latch for OUTPUT1 of OUTPUT-XBAR Writing 1 sets the corresponding output latch bit in the OUTPUTLATCH register Write of 0 has no effect Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.20 OUTPUTLATCHENABLE Register (Offset = 36h) [Reset = 00000000h]

OUTPUTLATCHENABLE is shown in Figure 9-77 and described in Table 9-85.

Return to the Summary Table.

Output X-BAR Output Latch Enable

Figure 9-77. OUTPUTLATCHENABLE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
OUTPUT8	OUTPUT7	OUTPUT6	OUTPUT5	OUTPUT4	OUTPUT3	OUTPUT2	OUTPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-85. OUTPUTLATCHENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	RESERVED	R-0	0h	Reserved
7	OUTPUT8	R/W	0h	Selects the output latch to drive OUTPUT8 for OUTPUT-XBAR 0: Output Latch is not selected to driven the respective output 1: Output Latch is selected to drive the respective output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	OUTPUT7	R/W	0h	Selects the output latch to drive OUTPUT7 for OUTPUT-XBAR 0: Output Latch is not selected to driven the respective output 1: Output Latch is selected to drive the respective output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	OUTPUT6	R/W	0h	Selects the output latch to drive OUTPUT6 for OUTPUT-XBAR 0: Output Latch is not selected to driven the respective output 1: Output Latch is selected to drive the respective output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	OUTPUT5	R/W	0h	Selects the output latch to drive OUTPUT5 for OUTPUT-XBAR 0: Output Latch is not selected to driven the respective output 1: Output Latch is selected to drive the respective output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	OUTPUT4	R/W	0h	Selects the output latch to drive OUTPUT4 for OUTPUT-XBAR 0: Output Latch is not selected to driven the respective output 1: Output Latch is selected to drive the respective output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	OUTPUT3	R/W	0h	Selects the output latch to drive OUTPUT3 for OUTPUT-XBAR 0: Output Latch is not selected to driven the respective output 1: Output Latch is selected to drive the respective output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-85. OUTPUTLATCHENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	OUTPUT2	R/W	0h	Selects the output latch to drive OUTPUT2 for OUTPUT-XBAR 0: Output Latch is not selected to driven the respective output 1: Output Latch is selected to drive the respective output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	OUTPUT1	R/W	0h	Selects the output latch to drive OUTPUT1 for OUTPUT-XBAR 0: Output Latch is not selected to driven the respective output 1: Output Latch is selected to drive the respective output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.21 OUTPUTINV Register (Offset = 38h) [Reset = 00000000h]

OUTPUTINV is shown in Figure 9-78 and described in Table 9-86.

Return to the Summary Table.

Output X-BAR Output Inversion

Figure 9-78. OUTPUTINV Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
OUTPUT8	OUTPUT7	OUTPUT6	OUTPUT5	OUTPUT4	OUTPUT3	OUTPUT2	OUTPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 9-86. OUTPUTINV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-8	RESERVED	R-0	0h	Reserved
7	OUTPUT8	R/W	0h	Selects polarity for OUTPUT8 of OUTPUT-XBAR 0: drives active high output 1: drives active-low output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
6	OUTPUT7	R/W	0h	Selects polarity for OUTPUT7 of OUTPUT-XBAR 0: drives active high output 1: drives active-low output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
5	OUTPUT6	R/W	0h	Selects polarity for OUTPUT6 of OUTPUT-XBAR 0: drives active high output 1: drives active-low output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
4	OUTPUT5	R/W	0h	Selects polarity for OUTPUT5 of OUTPUT-XBAR 0: drives active high output 1: drives active-low output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
3	OUTPUT4	R/W	0h	Selects polarity for OUTPUT4 of OUTPUT-XBAR 0: drives active high output 1: drives active-low output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
2	OUTPUT3	R/W	0h	Selects polarity for OUTPUT3 of OUTPUT-XBAR 0: drives active high output 1: drives active-low output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

Table 9-86. OUTPUTINV Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	OUTPUT2	R/W	0h	Selects polarity for OUTPUT2 of OUTPUT-XBAR 0: drives active high output 1: drives active-low output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn
0	OUTPUT1	R/W	0h	Selects polarity for OUTPUT1 of OUTPUT-XBAR 0: drives active high output 1: drives active-low output Refer to the Output X-BAR section of this chapter for more details. Reset type: CPU1.SYSRSn

9.3.5.22 OUTPUTLOCK Register (Offset = 3Eh) [Reset = 00000000h]

OUTPUTLOCK is shown in Figure 9-79 and described in Table 9-87.

Return to the Summary Table.

Output X-BAR Configuration Lock register

Figure 9-79. OUTPUTLOCK Register

31	30	29	28	27	26	25	24
KEY							
R-0/W1S-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W1S-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							LOCK
R-0-0h							R/WSonce-0h

Table 9-87. OUTPUTLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W1S	0h	Bit-0 of this register can be set only if KEY= 0x5a5a Reset type: CPU1.SYSRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/WSonce	0h	Locks the configuration for OUTPUT-XBAR. Once the configuration is locked, writes to the below registers for OUTPUT-XBAR is blocked. Registers Affected by the LOCK mechanism: OUTPUT-XBAROUTyMUX0TO15CFG OUTPUT-XBAROUTyMUX16TO31CFG OUTPUT-XBAROUTyMUXENABLE OUTPUT-XBAROUTLATENABLE OUTPUT-XBAROUTINV 0: Writes to the above registers are allowed 1: Writes to the above registers are blocked Note: [1] LOCK mechanism only apples to writes. Reads are never blocked. Reset type: CPU1.SYSRSn

Chapter 10

Direct Memory Access (DMA)

The direct memory access (DMA) module provides a hardware method of transferring data between peripherals and memory without intervention from the CPU; thereby, freeing up bandwidth for other system functions. Additionally, the DMA has the capability to orthogonally rearrange the data as the data is transferred as well as “ping-pong” data between buffers. These features are useful for structuring data into blocks for CPU processing.

10.1 Introduction

The strength of a controller is not measured purely in processor speed, but in total system capabilities. As a part of the equation, any time the CPU bandwidth for a given function can be reduced, the greater the system capabilities. Many times applications spend a significant amount of the bandwidth moving data, whether moving data from off-chip memory to on-chip memory, from a peripheral such as an analog-to-digital converter (ADC) to RAM, or from one peripheral to another. Furthermore, many times this data comes in a format that is not conducive to the optimum processing powers of the CPU. The DMA module described in this chapter has the ability to free up CPU bandwidth and rearrange the data into a pattern for more streamlined processing.

The DMA module is an event-based machine, meaning the DMA module requires a peripheral or software trigger to start a DMA transfer. Although the DMA module can be made into a periodic time-driven machine by configuring a timer as the DMA trigger source, there is no mechanism within the module to start memory transfers periodically. The DMA module has two independent DMA channels that can be configured separately and each channel contains an independent PIE interrupt to let the CPU know when a DMA transfer has either started or completed. Channel 1 has the ability to be configured at a higher priority than Channel 2. At the heart of the DMA is a state machine and tightly coupled address control logic. This address control logic allows for rearrangement of the block of data during the transfer as well as the process of ping-ponging data between buffers. Each of these features is discussed in detail in this chapter.

10.1.1 Features

DMA features include:

- Two channels with independent PIE interrupts
- Each DMA channel can be triggered from multiple peripheral trigger sources independently
- Word Size: 16-bit or 32-bit (SPI limited to 16-bit)
- Throughput: 3 cycles/word without arbitration

10.1.2 Block Diagram

Figure 10-1 shows a device-level block diagram of the DMA.

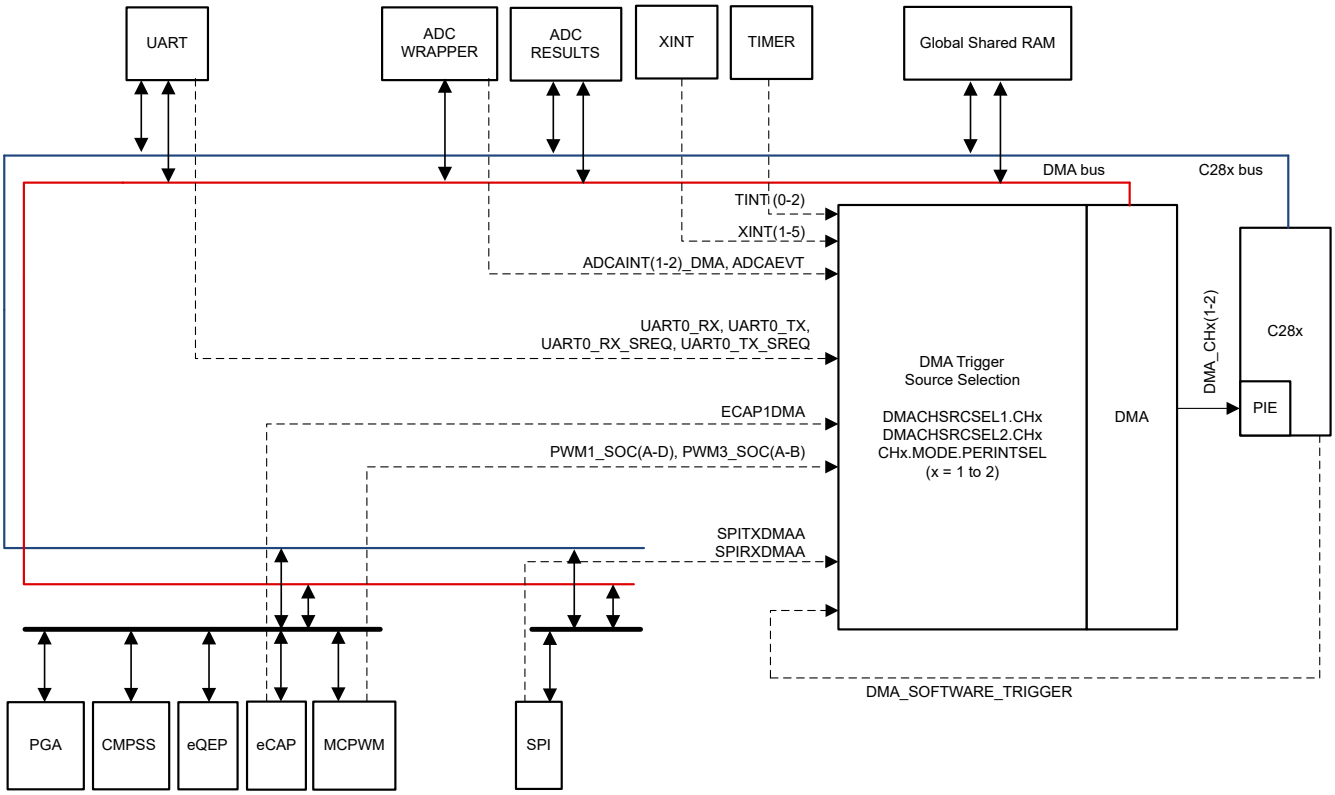


Figure 10-1. DMA Block Diagram

10.2 Architecture

10.2.1 Peripheral Interrupt Event Trigger Sources

Each DMA Channel can be configured to trigger by software and other peripheral triggers events. DMACHSRCSELx register can be used to configure DMA Trigger sources for each DMA channel. CHx.MODE.PERINTSEL register bit field can be set to channel number (CHx.MODE.PERINTSEL = x) as shown in Figure 10-2. Included in these DMA Trigger sources are five external interrupt signals that can be connected to most of the general-purpose input/output (GPIO) pins on the device. This adds significant flexibility to the event trigger capabilities. Upon receipt of a peripheral interrupt event signal, the DMA automatically sends a clear signal to the interrupt source so that subsequent interrupt events occur.

Note

To use the system-level DMA Trigger source selection, the DMA internal trigger source selection configuration for each channel can be done using the DMACHSRCSELx register and the CHx.MODE.PERINTSEL register. See Table 10-1 or the DMACHSRCSELx register definition for a complete list of DMA trigger sources.

Regardless of the value of the MODE.CHx[PERINTSEL] bit field, software can always force a trigger by using the CONTROL.CHx[PERINTFRC] bit. Likewise, software can always clear a pending DMA trigger using the CONTROL.CHx[PERINTCLR] bit.

Once a particular peripheral trigger event sets a channel's PERINTFLG bit, the bit remains pending until the priority logic of the state machine starts the burst transfer for that channel. Once the burst transfer starts, the flag is cleared. If a new peripheral trigger event is generated while a burst is in progress, the burst completes before responding to the new peripheral trigger event (after proper prioritization). If a third peripheral trigger event occurs before the pending event is serviced, an error flag is set in the CONTROL.CHx[OVRFLG] bit. If a peripheral trigger event occurs at the same time as the latched flag is being cleared, the trigger event has priority and the PERINTFLG remains set.

Figure 10-3 shows a diagram of the trigger select circuit.

Table 10-1 shows the peripheral trigger source options that are available for each channel.

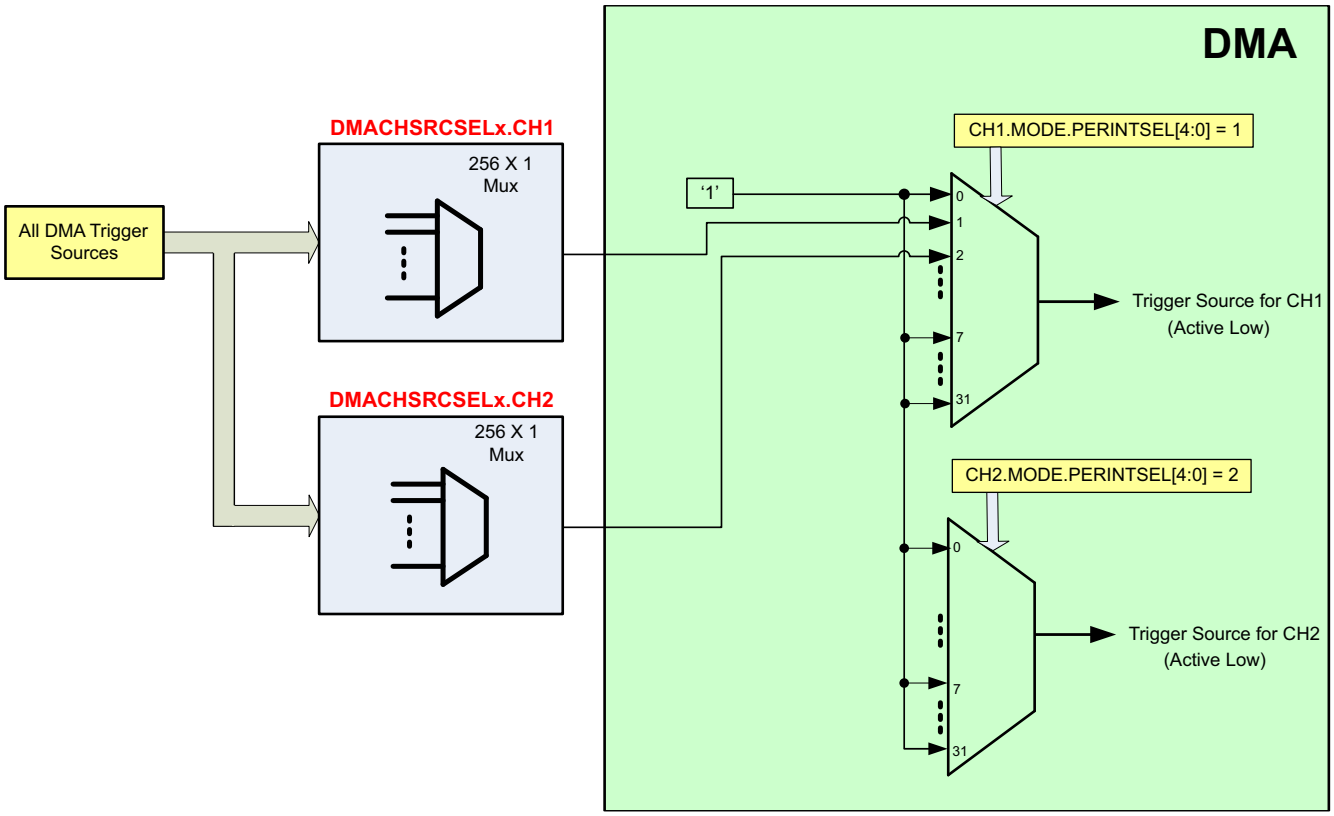
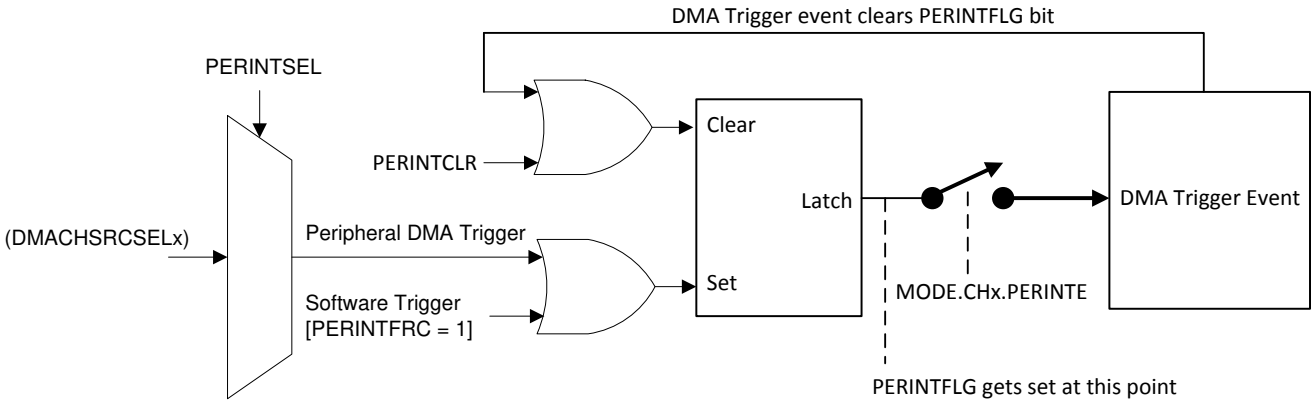


Figure 10-2. DMA Trigger Architecture



Note: See Figure 10-2.

Figure 10-3. Peripheral Interrupt Trigger Input Diagram

Table 10-1. DMA Trigger Source Options

Select Index	Trigger Source
0	DMA_SOFTWARE_TRIGGER
1	ADCAINT1_DMA
2	ADCAINT2_DMA
3	ADCAEVT
4	
5	

Table 10-1. DMA Trigger Source Options (continued)

Select Index	Trigger Source
6	
7	XINT1
8	XINT2
9	XINT3
10	XINT4
11	XINT5
12	PWM1_SOCA
13	PWM1_SOCB
14	PWM1_SOCC
15	PWM1_SOCD
16	
17	
18	
19	
20	PWM3_SOCA
21	PWM3_SOCB
22	
23	
24	
25	
26	
27	
28	TINT0
29	TINT1
30	TINT2
31	ECAP1DMA
32	
33	
34	
35	SPITXDMAA
36	SPIRXDMAA
37	UART0_RX
38	UART0_RX_SREQ
39	UART0_TX
40	UART0_TX_SREQ
41	
42	
43	
44-63	Reserved

10.2.2 DMA Bus

The DMA bus architecture consists of a 32-bit address bus, a 32-bit data read bus, and a 32-bit data write bus. Memories and register locations connected to the DMA bus by way of interfaces that sometimes share resources with the CPU memory or peripheral bus.

10.3 Address Pointer and Transfer Control

The DMA state machine is, at the most basic level, two nested loops.

Burst (Inner) Loop:

The burst (inner) loop transfers a programmable number of words set by (BURST_SIZE + 1) register when a DMA channel trigger (Peripheral or Software trigger) is received. The BURST_SIZE register allows a maximum of 32 sixteen-bit words to be transferred in one burst. Each DMA channel supports both 16-bit or 32-bit word burst that can be controlled by MODE.DATASIZE bit field. Each DMA channel contains a shadowed address pointer for the source (SRC_ADDR_SHADOW) and the destination (DST_ADDR_SHADOW) address. At the beginning of each transfer, the shadowed version of each pointer is copied into the respective active (SRC_ADDR_ACTIVE or DST_ADDR_ACTIVE) register. During the burst loop, after each word is transferred, the signed value contained in the appropriate source or destination BURST_STEP register is added to the active register:

SRC_ADDR_ACTIVE = SRC_ADDR_ACTIVE + SRC_BURST_STEP

DST_ADDR_ACTIVE = DST_ADDR_ACTIVE + DST_BURST_STEP

The burst (inner) loop transfers a burst of data when a DMA Channel Trigger (Peripheral or Software trigger) is received.

Transfer (Outer) Loop:

The Transfer (outer) loop transfers a programmable number of bursts set by (TRANSFER_SIZE + 1) register for each channel. Since TRANSFER_SIZE is a 16-bit register, the total size of a transfer allowed is well beyond any practical requirement. During the transfer loop, after each burst is complete, there are two methods that can be used to modify the active address pointer:

Method 1 (Default): When address wrapping is disabled (SRC_WRAP_SIZE or DST_WRAP_SIZE is greater than TRANSFER_SIZE), active address pointer is updated as shown below

SRC_ADDR_ACTIVE = SRC_ADDR_ACTIVE + SRC_TRANSFER_STEP

DST_ADDR_ACTIVE = DST_ADDR_ACTIVE + DST_TRANSFER_STEP

Method 2: Address wrapping gets enabled when SRC_WRAP_SIZE or DST_WRAP_SIZE is less than TRANSFER_SIZE. This allows the channel to wrap multiple times within a single transfer. When the number of bursts is equal to (SRC/DST_WRAP_SIZE + 1) register, the state machine modifies the active address pointers as:

SRC_BEG_ADDR_ACTIVE = SRC_BEG_ADDR_ACTIVE + SRC_WRAP_STEP

DST_BEG_ADDR_ACTIVE = DST_BEG_ADDR_ACTIVE + DST_WRAP_STEP

SRC_ADDR_ACTIVE = SRC_BEG_ADDR_ACTIVE

DST_ADDR_ACTIVE = DST_BEG_ADDR_ACTIVE

At the end of DMA transfer, DMA can have transferred (BURST_SIZE + 1) x (TRANSFER_SIZE + 1) words.

OneShot Mode:

OneShot mode is disabled by default.

When OneShot mode is disabled (MODE.CHx[ONESHOT] = 0), DMA transfers one burst [(BURST_SIZE + 1) words] of data each time a DMA Channel Trigger is received. After the burst is completed, the state machine moves on to the next pending channel in the priority scheme, even if another trigger for the channel just completed is pending. This feature keeps any single channel from monopolizing the DMA bus.

When OneShot mode is enabled (MODE.CHx[ONESHOT] = 1), DMA transfers all the bursts [(BURST_SIZE + 1) x (TRANSFER_SIZE + 1) words] on a single DMA channel trigger. Be careful when using this mode, since this can create a condition where one trigger uses up the majority of the DMA bandwidth.

Continuous Mode:

Continuous mode is disabled by default.

When Continuous mode is disabled (MODE.CHx[CONTINUOUS] = 0), DMA state machine disables channel after all bursts in a transfer loop (TRANSFER_COUNT = 0) are complete. The channel must be re-enabled by setting the RUN bit in the CONTROL register before another transfer can be started on that channel.

When Continuous mode is enabled (MODE.CHx[CONTINUOUS] = 1), DMA state machine keep channel active even after all bursts in a transfer loop (TRANSFER_COUNT = 0) are complete.

Each DMA channel can trigger an EPIE interrupt for each DMA transfer either at start of DMA transfer or end of DMA transfer using MODE.CHx[CHINTMODE] bit.

Source/Destination Address Pointers (SRC/DST_ADDR)	The value written into the shadow register is the start address of the first location where data is read or written to. At the beginning of a transfer the shadow register (SRC/DST_ADDR_SHADOW) is copied into the active register (SRC/DST_ADDR_ACTIVE). The active register performs as the current address pointer.
Source/Destination Begin Address Pointers (SRC/DST_BEG_ADDR)	This is the wrap pointer. The value written into the shadow register (SRC/DST_BEG_ADDR_SHADOW) is loaded into the active register (SRC/DST_BEG_ADDR_ACTIVE) at the start of a transfer. On a wrap condition, the active register (SRC/DST_BEG_ADDR_ACTIVE) is incremented by the signed value in the appropriate SRC/DST_WRAP_STEP register prior to being loaded into the active register (SRC/DST_ADDR_ACTIVE).

For each channel, the transfer process can be controlled with the following size values:

Source and Destination Burst Size (BURST_SIZE)

This specifies the number of words to be transferred in a burst.

This value is loaded into the BURST_COUNT register at the beginning of each burst. The BURST_COUNT decrements each word that is transferred and when the register reaches a zero value, the burst is complete, indicating that the next channel can be serviced. The behavior of the current channel is defined by the ONE_SHOT bit in the MODE register. The maximum size of the burst is dictated by the type of peripheral. For the ADC, the burst size can be all 16 registers (if all 16 registers are used). For RAM, the burst size can be up to the maximum allowed by the BURST_SIZE register, which is 32. See Table 10-2 to understand how BURST_SIZE register affects the number of 16-bit words transferred with respect to DATASIZE.

Table 10-2. BURSTSIZE versus DATASIZE Behavior

BURSTSIZE	Number of 16-bit words transferred in	
	DataSize = 16-bit data	DataSize = 32-bit data
0	1	2
1	2	2
2	3	4
3	4	4
4	5	6
5	6	6
6	7	8
7	8	8
8	9	10
9	10	10
10	11	12
11	12	12
*	*	*
*	*	*
*	*	*
30	31	32
31	32	32

Source and Destination Transfer Size (TRANSFER_SIZE)

This specifies the number of bursts to be transferred per CPU interrupt (if enabled).

Whether this interrupt is generated at the beginning or the end of the transfer is defined in the CHINTMODE bit in the MODE register. Whether the channel remains enabled or not after the transfer is completed is defined by the CONTINUOUS bit in the MODE register. The TRANSFER_SIZE register is loaded into the TRANSFER_COUNT register at the beginning of each transfer. The TRANSFER_COUNT register keeps track of how many bursts of data the channel has transferred and when the register reaches zero, the DMA transfer is complete.

Source/Destination Wrap Size (SRC/DST_WRAP_SIZE)

This specifies the number of bursts to be transferred before the current address pointer wraps around to the beginning.

This feature is used to implement a circular addressing type function. This value is loaded into the appropriate SRC/DST_WRAP_COUNT register at the beginning of each transfer. The SRC/DST_WRAP_COUNT registers keep track of how many bursts of data the channel has transferred and when the registers reach zero, the wrap procedure is performed on the appropriate source or destination address pointer. A separate size and count register is allocated for source and destination pointers. To disable the wrap function, assign the value of these registers to be larger than the TRANSFER_SIZE.

Note

The value written to the SIZE registers is one less than the intended size. So, to transfer three 16-bit words, the value 2 can be placed in the SIZE register.

Regardless of the state of the DATASIZE bit, the value specified in the SIZE registers are for 16-bit addresses. So, to transfer three 32-bit words, the value 5 can be placed in the SIZE register.

For each source/destination pointer, the address changes can be controlled with the following step values:

Source/Destination Burst Step (SRC/DST_BURST_STEP)

Within each burst transfer, the address source and destination step sizes are specified by these registers.

This value is a signed 2s compliment number so that the address pointer can be incremented or decremented as required. If no increment is desired, such as when accessing the data receive or transmit registers in a communication peripheral, the value of these registers can be set to zero.

Source/Destination Transfer Step (SRC/DST_TRANSFER_STEP)

This specifies the address offset to start the next burst transfer after completing the current burst transfer.

This is used in cases where registers or data memory locations are spaced at constant intervals. This value is a signed 2s compliment number so that the address pointer can be incremented or decremented as required.

Source/Destination Wrap Step (SRC/DST_WRAP_STEP)

When the wrap counter reaches zero, this value specifies the number of words to add/subtract from the SRC/DST_BEG_ADDR pointer and hence sets the new start address.

This implements a circular type of addressing mode, useful in many applications. This value is a signed 2s compliment number so that the address pointer can be incremented or decremented as required.

Note

Regardless of the state of the DATASIZE bit, the value specified in the STEP registers are for 16-bit addresses. So, to increment one 32-bit address, a value of 2 can be placed in these registers.

Channel Interrupt Mode (CHINTMODE)

This mode bit selects whether the DMA interrupt from the respective channel is generated at the beginning of a new transfer or at the end of the transfer.

If implementing a ping-pong buffer scheme with continuous mode of operation, then the interrupt can be generated at the beginning, just after the working registers are copied to the shadow set. If the DMA does not operate in continuous mode, then the interrupt is typically generated at the end when the transfer is complete.

All of the previous features and modes are shown in Figure 10-4. The following items are in reference to Figure 10-4.

- The *HALT* points represent where the channel halts operation when interrupted by a high priority channel 1 trigger, or when the HALT command is set, or when an emulation halt is issued and the FREE bit is cleared to 0.
- The SRC/DST_ADDR_ACTIVE registers are not affected by SRC/DST_BEG_ADDR_ACTIVE at the start of a transfer. SRC/DST_BEG_ADDR_ACTIVE only affects the SRC/DST_ADDR_ACTIVE registers on a wrap. Following is what happens when a transfer first starts:
  - SRC/DST_BEG_ADDR_SHADOW remains unchanged.
  - SRC/DST_ADDR_SHADOW remains unchanged.
  - SRC/DST_BEG_ADDR_ACTIVE = SRC/DST_BEG_ADDR_SHADOW
  - SRC/DST_ADDR_ACTIVE = SRC/DST_ADDR_SHADOW
- The active registers get updated when a wrap occurs. The shadow registers remain unchanged. Specifically:
  - SRC/DST_BEG_ADDR_SHADOW remains unchanged.
  - SRC/DST_ADDR_SHADOW remains unchanged.
  - SRC/DST_BEG_ADDR_ACTIVE += SRC/DST_WRAP_STEP
  - SRC/DST_ADDR_ACTIVE = SRC/DST_BEG_ADDR_ACTIVE
- The best way to remember this is:
  - The shadow registers never change except by software.
  - The active registers never change except by hardware, and a shadow register is only copied into the active register, never an active register by another name.

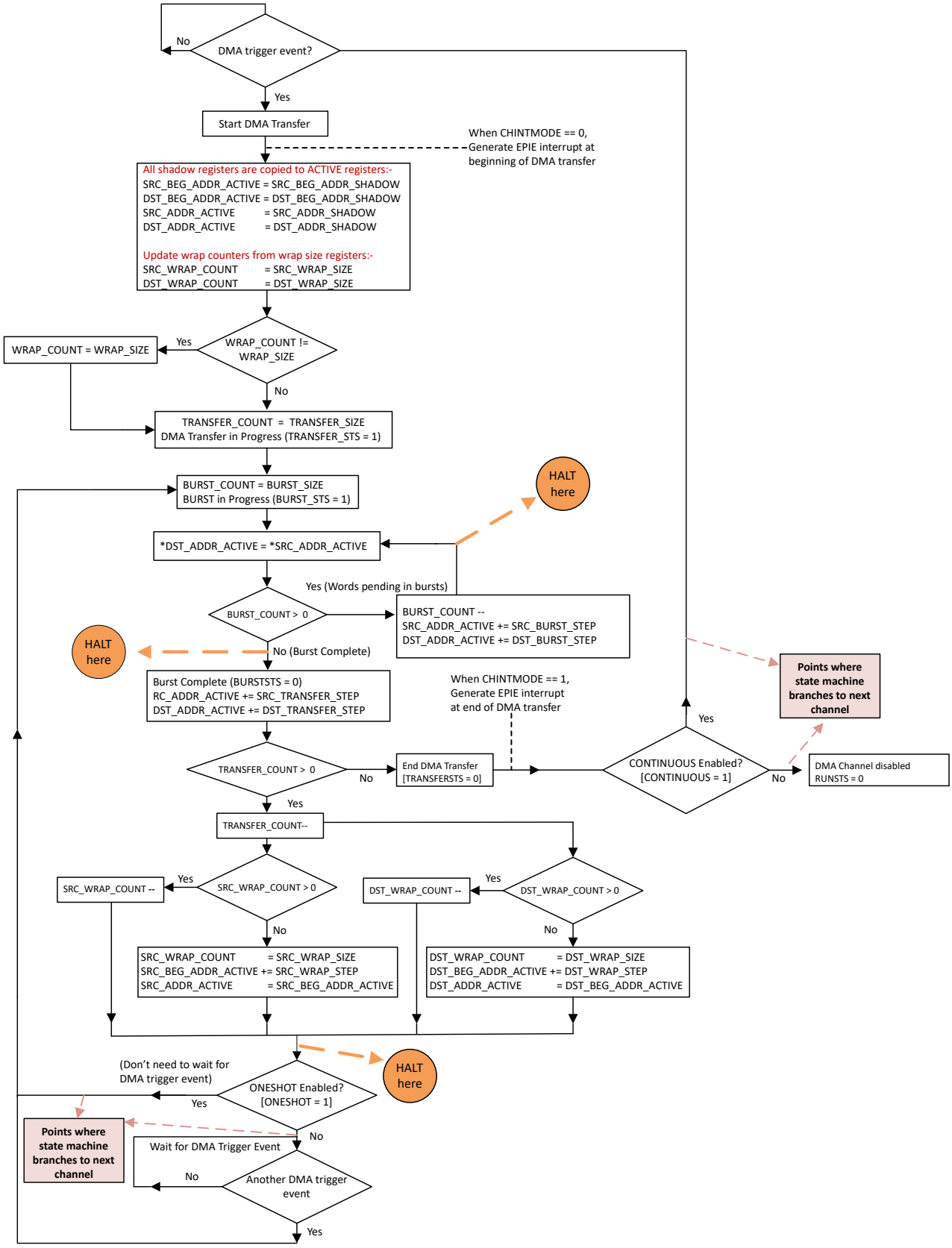


Figure 10-4. DMA State Diagram

10.4 Pipeline Timing and Throughput

In addition to the pipeline there are a few other behaviors of the DMA that affect the total throughput:

- A 1-cycle delay is added at the beginning of each burst
- A 1-cycle delay is added when returning from a CH1 high-priority interrupt
- Collisions with the CPU can add delay slots
- 32-bit transfers run at double the speed of a 16-bit transfer (takes the same amount of time to transfer a 32-bit word as to transfer a 16-bit word)

For example, to transfer 128 16-bit words from GS0 RAM to GS3 RAM, a channel can be configured to transfer 8 bursts of 16 words/burst. This gives:

8 bursts × [(3 cycles/word × 16 words/burst) + 1] = 392 cycles

If instead the channel were configured to transfer the same amount of data 32 bits at a time (the word size is configured to 32 bits), the transfer can take:

8 bursts × [(3 cycles/word × 8 words/burst) + 1] = 200 cycles

The DMA module consists of a 3-stage pipeline as shown in Figure 10-5 and Figure 10-6.

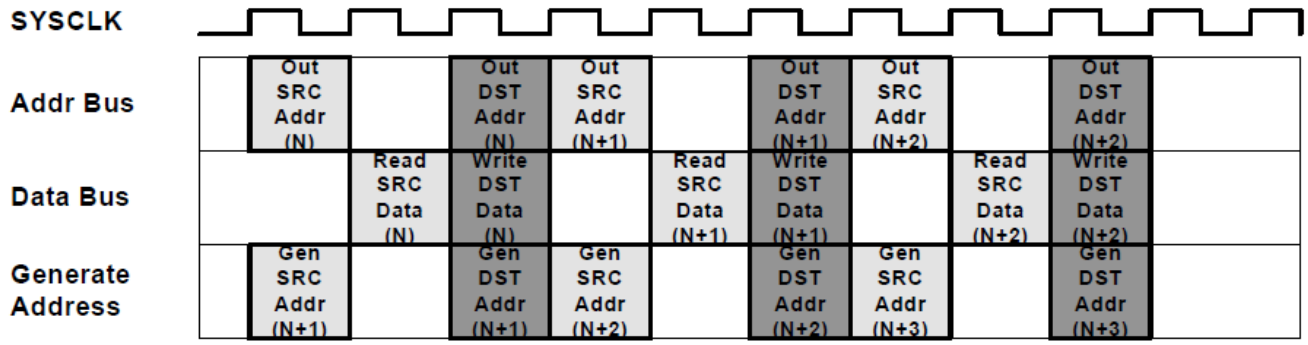


Figure 10-5. 3-Stage Pipeline DMA Transfer

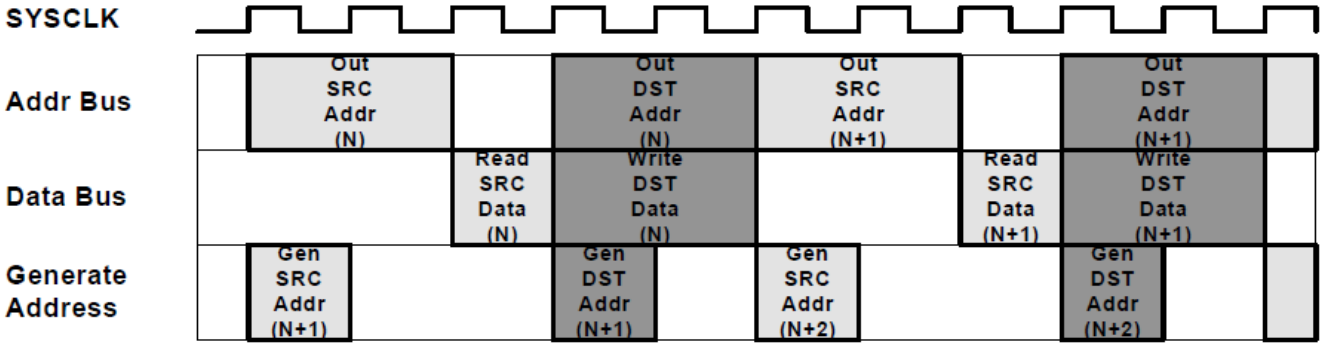


Figure 10-6. 3-stage Pipeline with One Read Stall

10.5 Channel Priority

Two priority schemes exist when determining channel priority: Round-robin mode and Channel 1 high-priority mode.

10.5.1 Round-Robin Mode

In this mode, both channels have *equal* priority and each enabled channel is serviced in round-robin fashion as:

CH1 → CH2 → CH1 → CH2 → CH1 …

In the case above, after each channel has transferred a burst of words, the next channel is serviced. The user can specify the size of the burst for each channel. Once (or the last enabled channel) has been serviced, and no other channels are pending, the round-robin state machine enters an idle state.

From the idle state, channel 1 (if enabled) is always serviced first. Both channels are of *equal* priority. For instance, take an example where both CH1 and CH2 are enabled in round-robin mode and CH2 is currently being processed. Then CH1 receives an interrupt trigger from the respective peripherals before CH2 completes. CH1 is now pending. When CH2 completes the *burst*, CH1 is serviced next. Only after CH2 completes is CH1 serviced. Upon completion of the CH1 *burst*, if there are no more channels pending, the round-robin state machine enters an idle state.

The round-robin state machine can be reset to the idle state using the DMACTRL[PRIORITYRESET] bit.

10.5.2 Channel 1 High-Priority Mode

In this mode, Channel 1 has high priority over Channel 2.

Higher priority:	CH1
Lower priority:	CH2

Given an example where both CH1 and CH2 are enabled in Channel 1 high-priority mode and CH2 is currently being processed. Then CH1 receives an interrupt trigger from the respective peripheral before CH2 completes. CH1 is now pending. When the current CH2 *word transfer* is completed, regardless of whether the DMA has completed the entire CH2 burst, CH2 execution is suspended and CH1 is serviced. After the CH1 burst completes, CH2 resumes execution.

Upon completion of the full bursts on both channels, there are no more channels pending, so the round-robin state machine enters an idle state.

Typically Channel 1 is used in this mode for the ADC, since the data rate is so high. However, Channel 1 high-priority mode can be used in conjunction with any peripheral.

Note

High-priority mode and ONESHOT mode cannot be used at the same time on Channel 1. Other channels can use ONESHOT mode when Channel 1 is in high-priority mode.

10.6 Overrun Detection Feature

The DMA contains overrun detection logic. When a peripheral event trigger is received by the DMA, the PERINTFLG bit in the CONTROL register is set, pending the channel to the DMA state machine. When the burst for that channel is started, the PERINTFLG is cleared. If however, between the time that the PERINTFLG bit is set by an event trigger and cleared by the start of the burst, an additional event trigger arrives, the second trigger is lost. This condition sets the OVRFLG bit in the CONTROL register as in Figure 10-7. If the overrun interrupt is enabled, the channel interrupt is generated to the PIE module.

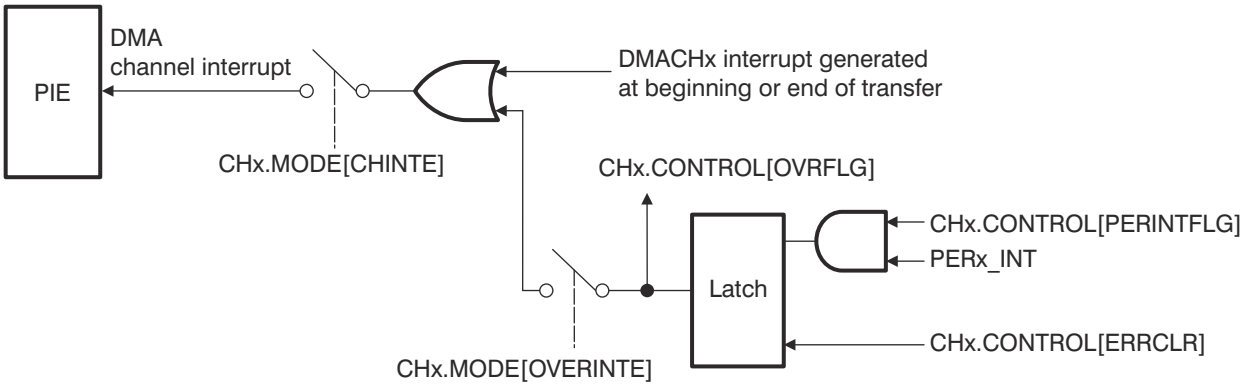


Figure 10-7. Overrun Detection Logic

10.7 Software

10.7.1 DMA Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/dma

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

10.7.1.1 DMA GSRAM Transfer (dma_ex1_gsram_transfer)

FILE: dma_ex1_gsram_transfer.c

This example uses one DMA channel to transfer data from a buffer in RAMGS0 to a buffer in RAMGS0. The example sets the DMA channel PERINTFRC bit repeatedly until the transfer of 16 bursts (where each burst is 8 16-bit words) has been completed. When the whole transfer is complete, it will trigger the DMA interrupt.

: This example project has support for migration across our C2000 device families. If you are wanting to build this project from launchpad or controlCARD, please specify in the .syscfg file the board you're using. At any time you can select another device to migrate this example. *Watch Variables*

- *sData* - Data to send
- *rData* - Received data

10.7.1.2 DMA GSRAM Transfer (dma_ex2_gsram_transfer)

FILE: dma_ex2_gsram_transfer.c

This example uses one DMA channel to transfer data from a buffer in RAMGS0 to a buffer in RAMGS0. The example sets the DMA channel PERINTFRC bit repeatedly until the transfer of 16 bursts (where each burst is 8 16-bit words) has been completed. When the whole transfer is complete, it will trigger the DMA interrupt.

Watch Variables

- *sData* - Data to send
- *rData* - Received data

10.8 DMA Registers

This Section describes the DMA Registers.

10.8.1 DMA Base Address Table

Table 10-3. DMA Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
DmaRegs	DMA_REGS	DMA_BASE	0x0000_1000	-	-
Dmach1Regs	DMA_CH_REGS	DMA_CH1_BASE	0x0000_1020	-	-
Dmach2Regs	DMA_CH_REGS	DMA_CH2_BASE	0x0000_1040	-	-

10.8.2 DMA_REGS Registers

Table 10-4 lists the memory-mapped registers for the DMA_REGS registers. All register offset addresses not listed in Table 10-4 should be considered as reserved locations and the register contents should not be modified.

Table 10-4. DMA_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	DMACTRL	DMA Control Register	EALLOW
1h	DEBUGCTRL	Debug Control Register	EALLOW
4h	PRIORITYCTRL1	Priority Control 1 Register	EALLOW
6h	PRIORITYSTAT	Priority Status Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 10-5 shows the codes that are used for access types in this section.

Table 10-5. DMA_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

10.8.2.1 DMACTRL Register (Offset = 0h) [Reset = 0000h]

DMACTRL is shown in Figure 10-8 and described in Table 10-6.

Return to the Summary Table.

DMA Control Register

Figure 10-8. DMACTRL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						PRIORITYRESET	HARDRESET
R-0h						R-0/W1S-0h	R-0/W1S-0h

Table 10-6. DMACTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-2	RESERVED	R	0h	Reserved
1	PRIORITYRESET	R-0/W1S	0h	The priority reset bit resets the round-robin state machine when a 1 is written. Service starts from the first enabled channel. Writes of 0 are ignored and this bit always reads back a 0. When a 1 is written to this bit, any pending burst transfer completes before resetting the channel priority machine. If CH1 is configured as a high-priority channel, and this bit is written to while CH1 is servicing a burst, both the CH1 burst and the next pending low-priority burst are completed before the state machine is reset. If CH1 is high-priority, the state machine restarts from CH2 (or the next highest enabled channel). Reset type: SYSRSn
0	HARDRESET	R-0/W1S	0h	Writing a 1 to the hard reset bit resets the whole DMA and aborts any current access (similar to applying a device reset). Writes of 0 are ignored and this bit always reads back a 0. For a soft reset, a bit is provided for each channel to perform a gentler reset. Refer to the channel control registers. When writing to this bit, there is a one cycle delay before it takes effect. Hence, a one-cycle delay (such as a NOP instruction) is required in software before attempting to access any other DMA register. Reset type: SYSRSn

10.8.2.2 DEBUGCTRL Register (Offset = 1h) [Reset = 0000h]

DEBUGCTRL is shown in Figure 10-9 and described in Table 10-7.

Return to the Summary Table.

Debug Control Register

Figure 10-9. DEBUGCTRL Register

15	14	13	12	11	10	9	8
FREE	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 10-7. DEBUGCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FREE	R/W	0h	Emulation Control This bit specifies the action when an emulation halt event occurs. Reset type: SYSRSn 0h (R/W) = The DMA completes the current read-write operation, then halts. 1h (R/W) = The DMA continues running during an emulation halt.
14-0	RESERVED	R	0h	Reserved

10.8.2.3 PRIORITYCTRL1 Register (Offset = 4h) [Reset = 0000h]

PRIORITYCTRL1 is shown in Figure 10-10 and described in Table 10-8.

Return to the Summary Table.

Priority Control 1 Register

Figure 10-10. PRIORITYCTRL1 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CH1PRIORITY
R-0h							R/W-0h

Table 10-8. PRIORITYCTRL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R	0h	Reserved
0	CH1PRIORITY	R/W	0h	DMA Channel 1 Priority This bit selects whether CH1 has high priority or not. The priority can only be changed when all channels are disabled. A priority reset should be performed before restarting channels after changing priority Reset type: SYSRSn 0h (R/W) = CH1 has the same priority as the other channels 1h (R/W) = CH1 has a higher priority than the other channels

10.8.2.4 PRIORITYSTAT Register (Offset = 6h) [Reset = 0000h]

PRIORITYSTAT is shown in Figure 10-11 and described in Table 10-9.

Return to the Summary Table.

Priority Status Register

Figure 10-11. PRIORITYSTAT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	ACTIVESTS_SHADOW			RESERVED	ACTIVESTS		
R-0h	R-0h			R-0h	R-0h		

Table 10-9. PRIORITYSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-4	ACTIVESTS_SHADOW	R	0h	Active Channel Status Shadow These bits are only meaningful when CH1 is in high-priority mode. When CH1 is serviced, the ACTIVESTS bits are copied to the shadow bits and indicate which channel was interrupted by CH1. When CH1 service is completed, the shadow bits are copied back to the ACTIVESTS bits. If this bit field is zero or the same as the ACTIVESTS bit field, then no channel is pending due to a CH1 interrupt. When CH1 is not a higher priority channel, these bits should be ignored. Reset type: SYSRSn 0h (R/W) = No channel is active 1h (R/W) = CH 1 2h (R/W) = CH 2 3h (R/W) = Reserved 4h (R/W) = Reserved 5h (R/W) = Reserved 6h (R/W) = Reserved 7h (R/W) = Reserved
3	RESERVED	R	0h	Reserved
2-0	ACTIVESTS	R	0h	Active Channel Status These bits indicate which channel (if any) is currently active or performing a transfer. Reset type: SYSRSn 0h (R/W) = No channel is active 1h (R/W) = CH 1 2h (R/W) = CH 2 3h (R/W) = Reserved 4h (R/W) = Reserved 5h (R/W) = Reserved 6h (R/W) = Reserved 7h (R/W) = Reserved

10.8.3 DMA_CH_REGS Registers

Table 10-10 lists the memory-mapped registers for the DMA_CH_REGS registers. All register offset addresses not listed in Table 10-10 should be considered as reserved locations and the register contents should not be modified.

Table 10-10. DMA_CH_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	MODE	Mode Register	EALLOW
1h	CONTROL	Control Register	EALLOW
2h	BURST_SIZE	Burst Size Register	EALLOW
3h	BURST_COUNT	Burst Count Register	EALLOW
4h	SRC_BURST_STEP	Source Burst Step Register	EALLOW
5h	DST_BURST_STEP	Destination Burst Step Register	EALLOW
6h	TRANSFER_SIZE	Transfer Size Register	EALLOW
7h	TRANSFER_COUNT	Transfer Count Register	EALLOW
8h	SRC_TRANSFER_STEP	Source Transfer Step Register	EALLOW
9h	DST_TRANSFER_STEP	Destination Transfer Step Register	EALLOW
Ah	SRC_WRAP_SIZE	Source Wrap Size Register	EALLOW
Bh	SRC_WRAP_COUNT	Source Wrap Count Register	EALLOW
Ch	SRC_WRAP_STEP	Source Wrap Step Register	EALLOW
Dh	DST_WRAP_SIZE	Destination Wrap Size Register	EALLOW
Eh	DST_WRAP_COUNT	Destination Wrap Count Register	EALLOW
Fh	DST_WRAP_STEP	Destination Wrap Step Register	EALLOW
10h	SRC_BEG_ADDR_SHADOW	Source Begin Address Shadow Register	EALLOW
12h	SRC_ADDR_SHADOW	Source Address Shadow Register	EALLOW
14h	SRC_BEG_ADDR_ACTIVE	Source Begin Address Active Register	EALLOW
16h	SRC_ADDR_ACTIVE	Source Address Active Register	EALLOW
18h	DST_BEG_ADDR_SHADOW	Destination Begin Address Shadow Register	EALLOW
1Ah	DST_ADDR_SHADOW	Destination Address Shadow Register	EALLOW
1Ch	DST_BEG_ADDR_ACTIVE	Destination Begin Address Active Register	EALLOW
1Eh	DST_ADDR_ACTIVE	Destination Address Active Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 10-11 shows the codes that are used for access types in this section.

Table 10-11. DMA_CH_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

10.8.3.1 MODE Register (Offset = 0h) [Reset = 0000h]

MODE is shown in Figure 10-12 and described in Table 10-12.

Return to the Summary Table.

Mode Register

Figure 10-12. MODE Register

15	14	13	12	11	10	9	8
CHINTE	DATASIZE	RESERVED	RESERVED	CONTINUOUS	ONESHOT	CHINTMODE	PERINTE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
OVRINTE	RESERVED		PERINTSEL				
R/W-0h	R-0h		R/W-0h				

Table 10-12. MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CHINTE	R/W	0h	Channel Interrupt Enable Bit This bit enables the DMA channel's CPU interrupt. Reset type: SYSRSn 0h (R/W) = Interrupt disabled 1h (R/W) = Interrupt enabled
14	DATASIZE	R/W	0h	Data Size Mode Bit This bit determines whether the DMA channel transfers 16 bits or 32 bits of data per read/write operation. Regardless of this setting, all data lengths and offsets in other DMA registers refer to 16- bit words. The pointer step increments must be configured to accomodate 32-bit words. Reset type: SYSRSn 0h (R/W) = 16-bit data transfer size 1h (R/W) = 32-bit data transfer size
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	CONTINUOUS	R/W	0h	Continuous Mode Bit If this bit is set to 1, then the channel re-initializes when TRANSFER_COUNT is zero and waits for the next event trigger. Otherwise, the DMA stops and clears the RUNSTS bit. Reset type: SYSRSn
10	ONESHOT	R/W	0h	One Shot Mode If this bit is set to 1, each peripheral event trigger causes the channel to perform an entire transfer. Otherwise, the channel only performs one burst per trigger. Reset type: SYSRSn
9	CHINTMODE	R/W	0h	Channel Interrupt Generation Mode This bit specifies when the DMA channel generates a CPU interrupt for a transfer. Reset type: SYSRSn 0h (R/W) = Generate interrupt at beginning of new transfer 1h (R/W) = Generate interrupt at end of transfer.
8	PERINTE	R/W	0h	Peripheral Event Trigger Enable This bit enables peripheral event triggers on the DMA channel. Reset type: SYSRSn 0h (R/W) = Peripheral event trigger disabled. Neither the selected peripheral nor software can start a DMA burst. 1h (R/W) = Peripheral event trigger enabled.

Table 10-12. MODE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	OVRINTE	R/W	0h	Overflow Interrupt Enable The bit determines whether the DMA module generates a CPU interrupt when it detects an overflow event. Reset type: SYSRSn 0h (R/W) = Overflow interrupt disabled 1h (R/W) = Overflow interrupt enabled
6-5	RESERVED	R	0h	Reserved
4-0	PERINTSEL	R/W	0h	Peripheral Event Trigger Source Select These are legacy bits and should be set to the channel number. The actual source selection is done via the DMACHSRCSELn registers, which are part of the DMA_CLA_SRC_SEL_REGS group. Reset type: SYSRSn

10.8.3.2 CONTROL Register (Offset = 1h) [Reset = 0000h]

CONTROL is shown in Figure 10-13 and described in Table 10-13.

Return to the Summary Table.

Control Register

Figure 10-13. CONTROL Register

15	14	13	12	11	10	9	8
RESERVED	OVRFLG	RUNSTS	BURSTSTS	TRANSFERSTS	RESERVED	RESERVED	PERINTFLG
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
ERRCLR	RESERVED	RESERVED	PERINTCLR	PERINTFRC	SOFTRESET	HALT	RUN
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 10-13. CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	OVRFLG	R	0h	Overflow Flag This bit indicates that a peripheral event trigger was received while PERINTFLG was already set. It can be cleared by writing to the ERRCLR bit. Reset type: SYSRSn 0h (R/W) = No overflow detected 1h (R/W) = Overflow detected
13	RUNSTS	R	0h	Run Status Flag This bit indicates that the DMA channel is ready to respond to peripheral event triggers. This bit is set when a 1 is written to the RUN bit. It is cleared when a transfer completes (TRANSFER_COUNT = 0) and continuous mode is disabled, or when the HARDRESET, SOFTRESET, or HALT bit is set. Reset type: SYSRSn 0h (R/W) = The channel is disabled 1h (R/W) = The channel is enabled
12	BURSTSTS	R	0h	Burst Status Flag This bit is set when a DMA burst begins. The BURST_COUNT is set to the BURST_SIZE. This bit is cleared when BURST_COUNT reaches zero, or when the HARDRESET or SOFTRESET bit is set. Reset type: SYSRSn 0h (R/W) = No burst activity 1h (R/W) = The DMA is currently servicing or suspending a burst transfer from this channel
11	TRANSFERSTS	R	0h	Transfer Status Flag This bit is set when a DMA transfer begins. The address registers are copied to the shadow set and the TRANSFER_COUNT is set to the TRANSFER_SIZE. This bit is cleared when TRANSFER_COUNT reaches zero, or when the HARDRESET or SOFTRESET bit is set. Reset type: SYSRSn 0h (R/W) = No transfer activity 1h (R/W) = The channel is currently in the middle of a transfer regardless of whether a burst of data is actively being transferred or not
10	RESERVED	R	0h	Reserved
9	RESERVED	R	0h	Reserved

Table 10-13. CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	PERINTFLG	R	0h	Peripheral Event Trigger Flag This bit indicates whether a peripheral event trigger has arrived. This bit is automatically cleared when the first burst transfer begins. Reset type: SYSRSn 0h (R/W) = Waiting for event trigger 1h (R/W) = Event trigger pending
7	ERRCLR	R-0/W1S	0h	Clear Error Writing a 1 to this bit will clear the OVRFLG bit. This is normally done when initializing the DMA module or if an overflow condition is detected. If an overflow event occurs at the same time this bit is set, the overrun has priority and the OVRFLG bit is set. Reset type: SYSRSn
6	RESERVED	R-0/W1S	0h	Reserved
5	RESERVED	R-0/W1S	0h	Reserved
4	PERINTCLR	R-0/W1S	0h	Clear Peripheral Event Trigger Writing a 1 to this bit clears PERINTFLG, which cancels a pending event trigger. This is normally done when initializing the DMA module. If an event trigger arrives at the same time this bit is set, the trigger has priority and PERINTFLG is set. Reset type: SYSRSn
3	PERINTFRC	R-0/W1S	0h	Force Peripheral Event Trigger If the PERINTE bit of the MODE register is set, writing a 1 to this bit sets PERINTFLG, which triggers a DMA burst. This bit can be used to start a DMA transfer in software. Reset type: SYSRSn
2	SOFTRESET	R-0/W1S	0h	Channel Soft Reset Writing a 1 to this bit places the channel into its default state after the current read/write access has completed: RUNSTS = 0 TRANSFERSTS = 0 BURSTSTS = 0 BURST_COUNT = 0 TRANSFER_COUNT = 0 SRC_WRAP_COUNT = 0 DST_WRAP_COUNT = 0 When writing to this bit, there is a one cycle delay before it takes effect. Hence, a one-cycle delay (such as a NOP instruction) is required in software before attempting to access any other DMA register. Reset type: SYSRSn
1	HALT	R-0/W1S	0h	Halt Channel Writing a 1 to this bit halts the DMA channel in its current state after any ongoing read/write access has completed. Reset type: SYSRSn
0	RUN	R-0/W1S	0h	Run Channel Writing a 1 to this bit enables the DMA channel and sets the RUNSTS bit to 1. This bit is also used to resume after a channel halt. The RUN bit is typically used to start the DMA channel after configuration. The channel will then wait for the first peripheral event trigger (PERINTFLG == 1) to start a burst. Reset type: SYSRSn

10.8.3.3 BURST_SIZE Register (Offset = 2h) [Reset = 0000h]

BURST_SIZE is shown in Figure 10-14 and described in Table 10-14.

Return to the Summary Table.

Burst Size Register

Figure 10-14. BURST_SIZE Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			BURSTSIZE				
R-0h			R/W-0h				

Table 10-14. BURST_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4-0	BURSTSIZE	R/W	0h	These bits specify the burst size in 16-bit words. The actual size is equal to BURSTSIZE + 1. Reset type: SYSRSn

10.8.3.4 BURST_COUNT Register (Offset = 3h) [Reset = 0000h]

BURST_COUNT is shown in Figure 10-15 and described in Table 10-15.

Return to the Summary Table.

Burst Count Register

Figure 10-15. BURST_COUNT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			BURSTCOUNT				
R-0h			R-0h				

Table 10-15. BURST_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4-0	BURSTCOUNT	R	0h	These bits indicate the number of words left in the current burst. Reset type: SYSRSn 0h (R/W) = 0 word left in a burst 1h (R/W) = 1 word left in a burst 2h (R/W) = 2 word left in a burst 3h (R/W) = 3 word left in a burst 4h (R/W) = 4 word left in a burst 5h (R/W) = 5 word left in a burst 6h (R/W) = 6 word left in a burst 7h (R/W) = 7 word left in a burst 8h (R/W) = 8 word left in a burst 9h (R/W) = 9 word left in a burst Ah (R/W) = 10 word left in a burst Bh (R/W) = 11 word left in a burst Ch (R/W) = 12 word left in a burst Dh (R/W) = 13 word left in a burst Eh (R/W) = 14 word left in a burst Fh (R/W) = 15 word left in a burst 10h (R/W) = 16 word left in a burst 11h (R/W) = 17 word left in a burst 12h (R/W) = 18 word left in a burst 13h (R/W) = 19 word left in a burst 14h (R/W) = 20 word left in a burst 15h (R/W) = 21 word left in a burst 16h (R/W) = 22 word left in a burst 17h (R/W) = 23 word left in a burst 18h (R/W) = 24 word left in a burst 19h (R/W) = 25 word left in a burst 1Ah (R/W) = 26 word left in a burst 1Bh (R/W) = 27 word left in a burst 1Ch (R/W) = 28 word left in a burst 1Dh (R/W) = 29 word left in a burst 1Eh (R/W) = 30 word left in a burst 1Fh (R/W) = 31 word left in a burst

10.8.3.5 SRC_BURST_STEP Register (Offset = 4h) [Reset = 0000h]

SRC_BURST_STEP is shown in Figure 10-16 and described in Table 10-16.

Return to the Summary Table.

Source Burst Step Register

Figure 10-16. SRC_BURST_STEP Register

15	14	13	12	11	10	9	8
SRCBURSTSTEP							
R/W-0h							
7	6	5	4	3	2	1	0
SRCBURSTSTEP							
R/W-0h							

Table 10-16. SRC_BURST_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	SRCBURSTSTEP	R/W	0h	These bits specify the change in the source address after each word in a burst. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the source address after each read/write operation in a burst. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

10.8.3.6 DST_BURST_STEP Register (Offset = 5h) [Reset = 0000h]

DST_BURST_STEP is shown in Figure 10-17 and described in Table 10-17.

Return to the Summary Table.

Destination Burst Step Register

Figure 10-17. DST_BURST_STEP Register

15	14	13	12	11	10	9	8
DSTBURSTSTEP							
R/W-0h							
7	6	5	4	3	2	1	0
DSTBURSTSTEP							
R/W-0h							

Table 10-17. DST_BURST_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DSTBURSTSTEP	R/W	0h	These bits specify the change in the destination address after each word in a burst. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the destination address after each read/write operation in a burst. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

10.8.3.7 TRANSFER_SIZE Register (Offset = 6h) [Reset = 0000h]

TRANSFER_SIZE is shown in Figure 10-18 and described in Table 10-18.

Return to the Summary Table.

Transfer Size Register

Figure 10-18. TRANSFER_SIZE Register

15	14	13	12	11	10	9	8
RESERVED							TRANSFERSIZE
R-0h							R/W-0h
7	6	5	4	3	2	1	0
TRANSFERSIZE							
R/W-0h							

Table 10-18. TRANSFER_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8-0	TRANSFERSIZE	R/W	0h	These bits specify the transfer size in bursts. The actual size is equal to TRANSFERSIZE + 1. Reset type: SYSRSn

10.8.3.8 TRANSFER_COUNT Register (Offset = 7h) [Reset = 0000h]

TRANSFER_COUNT is shown in Figure 10-19 and described in Table 10-19.

Return to the Summary Table.

Transfer Count Register

Figure 10-19. TRANSFER_COUNT Register

15	14	13	12	11	10	9	8
RESERVED							TRANSFERCOUNT
R-0h							R-0h
7	6	5	4	3	2	1	0
TRANSFERCOUNT							
R-0h							

Table 10-19. TRANSFER_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8-0	TRANSFERCOUNT	R	0h	These bits indicate the number of bursts left in the current transfer. Reset type: SYSRSn

10.8.3.9 SRC_TRANSFER_STEP Register (Offset = 8h) [Reset = 0000h]

SRC_TRANSFER_STEP is shown in Figure 10-20 and described in Table 10-20.

Return to the Summary Table.

Source Transfer Step Register

Figure 10-20. SRC_TRANSFER_STEP Register

15	14	13	12	11	10	9	8
SRCTRANSFERSTEP							
R/W-0h							
7	6	5	4	3	2	1	0
SRCTRANSFERSTEP							
R/W-0h							

Table 10-20. SRC_TRANSFER_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	SRCTRANSFERSTEP	R/W	0h	These bits specify the change in the source address after a burst completes. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the source address after each burst completes. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

10.8.3.10 DST_TRANSFER_STEP Register (Offset = 9h) [Reset = 0000h]

DST_TRANSFER_STEP is shown in Figure 10-21 and described in Table 10-21.

Return to the Summary Table.

Destination Transfer Step Register

Figure 10-21. DST_TRANSFER_STEP Register

15	14	13	12	11	10	9	8
DSTTRANSFERSTEP							
R/W-0h							
7	6	5	4	3	2	1	0
DSTTRANSFERSTEP							
R/W-0h							

Table 10-21. DST_TRANSFER_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DSTTRANSFERSTEP	R/W	0h	These bits specify the change in the destination address after a burst completes. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the destination address after each burst completes. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

10.8.3.11 SRC_WRAP_SIZE Register (Offset = Ah) [Reset = 00FFh]

SRC_WRAP_SIZE is shown in Figure 10-22 and described in Table 10-22.

Return to the Summary Table.

Source Wrap Size Register

Figure 10-22. SRC_WRAP_SIZE Register

15	14	13	12	11	10	9	8
RESERVED							WRAPSIZE
R-0h							R/W-FFh
7	6	5	4	3	2	1	0
WRAPSIZE							
R/W-FFh							

Table 10-22. SRC_WRAP_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8-0	WRAPSIZE	R/W	FFh	These bits specify the number of bursts to transfer before the source address wraps around to the beginning address. The actual number is equal to WRAPSIZE + 1. To disable the wrapping function, set WRAPSIZE to a value larger than TRANSFERSIZE. Reset type: SYSRSn

10.8.3.12 SRC_WRAP_COUNT Register (Offset = Bh) [Reset = 0000h]

SRC_WRAP_COUNT is shown in Figure 10-23 and described in Table 10-23.

Return to the Summary Table.

Source Wrap Count Register

Figure 10-23. SRC_WRAP_COUNT Register

15	14	13	12	11	10	9	8
RESERVED							WRAPSIZE
R-0h							R-0h
7	6	5	4	3	2	1	0
WRAPSIZE							
R-0h							

Table 10-23. SRC_WRAP_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8-0	WRAPSIZE	R	0h	These bits indicate the number of bursts left before wrapping the source address. Reset type: SYSRSn

10.8.3.13 SRC_WRAP_STEP Register (Offset = Ch) [Reset = 0000h]

SRC_WRAP_STEP is shown in Figure 10-24 and described in Table 10-24.

Return to the Summary Table.

Source Wrap Step Register

Figure 10-24. SRC_WRAP_STEP Register

15	14	13	12	11	10	9	8
WRAPSTEP							
R/W-0h							
7	6	5	4	3	2	1	0
WRAPSTEP							
R/W-0h							

Table 10-24. SRC_WRAP_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	WRAPSTEP	R/W	0h	These bits specify the change in the source beginning address when the wrap counter reaches zero. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the source address when wrapping occurs. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

10.8.3.14 DST_WRAP_SIZE Register (Offset = Dh) [Reset = 00FFh]

DST_WRAP_SIZE is shown in Figure 10-25 and described in Table 10-25.

Return to the Summary Table.

Destination Wrap Size Register

Figure 10-25. DST_WRAP_SIZE Register

15	14	13	12	11	10	9	8
RESERVED							WRAPSIZE
R-0h							R/W-FFh
7	6	5	4	3	2	1	0
WRAPSIZE							
R/W-FFh							

Table 10-25. DST_WRAP_SIZE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8-0	WRAPSIZE	R/W	FFh	These bits specify the number of bursts to transfer before the destination address wraps around to the beginning address. The actual number is equal to WRAPSIZE + 1. To disable the wrapping function, set WRAPSIZE to a value larger than TRANSFERSIZE. Reset type: SYSRSn

10.8.3.15 DST_WRAP_COUNT Register (Offset = Eh) [Reset = 0000h]

DST_WRAP_COUNT is shown in Figure 10-26 and described in Table 10-26.

Return to the Summary Table.

Destination Wrap Count Register

Figure 10-26. DST_WRAP_COUNT Register

15	14	13	12	11	10	9	8
RESERVED							WRAPSIZE
R-0h							R-0h
7	6	5	4	3	2	1	0
WRAPSIZE							
R-0h							

Table 10-26. DST_WRAP_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8-0	WRAPSIZE	R	0h	These bits indicate the number of bursts left before wrapping the destination address. Reset type: SYSRSn

10.8.3.16 DST_WRAP_STEP Register (Offset = Fh) [Reset = 0000h]

DST_WRAP_STEP is shown in Figure 10-27 and described in Table 10-27.

Return to the Summary Table.

Destination Wrap Step Register

Figure 10-27. DST_WRAP_STEP Register

15	14	13	12	11	10	9	8
WRAPSTEP							
R/W-0h							
7	6	5	4	3	2	1	0
WRAPSTEP							
R/W-0h							

Table 10-27. DST_WRAP_STEP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	WRAPSTEP	R/W	0h	These bits specify the change in the destination beginning address when the wrap counter reaches zero. The size must be a 16-bit two's complement value between -4096 and 4095 (inclusive). This value is added to the destination address when wrapping occurs. Reset type: SYSRSn 0h (R/W) = No address change 1h (R/W) = Add 1 to the address 2h (R/W) = Add 2 to the address FFEh (R/W) = Add 4094 to the address FFFh (R/W) = Add 4095 to the address F000h (R/W) = Subtract 4096 from the address F001h (R/W) = Subtract 4095 from the address FFFEh (R/W) = Subtract 2 from the address FFFFh (R/W) = Subtract 1 from the address

10.8.3.17 SRC_BEG_ADDR_SHADOW Register (Offset = 10h) [Reset = 00000000h]

SRC_BEG_ADDR_SHADOW is shown in Figure 10-28 and described in Table 10-28.

Return to the Summary Table.

Source Begin Address Shadow Register

Figure 10-28. SRC_BEG_ADDR_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEGADDR																															
R/W-0h																															

Table 10-28. SRC_BEG_ADDR_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BEGADDR	R/W	0h	Shadow Source Beginning Address At the start of a transfer, the value in this register is loaded into the SRC_BEG_ADDR_ACTIVE register and used as the beginning value for the source address. This register can be safely updated during a transfer. Reset type: SYSRSn

10.8.3.18 SRC_ADDR_SHADOW Register (Offset = 12h) [Reset = 00000000h]

SRC_ADDR_SHADOW is shown in Figure 10-29 and described in Table 10-29.

Return to the Summary Table.

Source Address Shadow Register

Figure 10-29. SRC_ADDR_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
R/W-0h																															

Table 10-29. SRC_ADDR_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Shadow Source Address At the start of a transfer, the value in this register is loaded into the SRC_ADDR_ACTIVE register and used as the value of the source address. This register can be safely updated during a transfer. Reset type: SYSRSn

10.8.3.19 SRC_BEG_ADDR_ACTIVE Register (Offset = 14h) [Reset = 00000000h]

SRC_BEG_ADDR_ACTIVE is shown in Figure 10-30 and described in Table 10-30.

Return to the Summary Table.

Source Begin Address Active Register

Figure 10-30. SRC_BEG_ADDR_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEGADDR																															
R-0h																															

Table 10-30. SRC_BEG_ADDR_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BEGADDR	R	0h	Active Source Beginning Address If a transfer is ongoing, this register holds the current beginning value for the source address. This address may be updated after wrapping. When a transfer starts, this register is loaded with the shadow address from the SRC_BEG_ADDR_SHADOW register. Reset type: SYSRSn

10.8.3.20 SRC_ADDR_ACTIVE Register (Offset = 16h) [Reset = 00000000h]

SRC_ADDR_ACTIVE is shown in Figure 10-31 and described in Table 10-31.

Return to the Summary Table.

Source Address Active Register

Figure 10-31. SRC_ADDR_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
R-0h																															

Table 10-31. SRC_ADDR_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	Active Source Address If a transfer is ongoing, this register holds the current value of the source address. This address may change after a write, a burst, or wrapping. Reset type: SYSRSn

10.8.3.21 DST_BEG_ADDR_SHADOW Register (Offset = 18h) [Reset = 00000000h]

DST_BEG_ADDR_SHADOW is shown in Figure 10-32 and described in Table 10-32.

Return to the Summary Table.

Destination Begin Address Shadow Register

Figure 10-32. DST_BEG_ADDR_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEGADDR																															
R/W-0h																															

Table 10-32. DST_BEG_ADDR_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BEGADDR	R/W	0h	Shadow Destination Beginning Address At the start of a transfer, the value in this register is loaded into the DST_BEG_ADDR_ACTIVE register and used as the beginning value for the destination address. This register can be safely updated during a transfer. Reset type: SYSRSn

10.8.3.22 DST_ADDR_SHADOW Register (Offset = 1Ah) [Reset = 00000000h]

DST_ADDR_SHADOW is shown in Figure 10-33 and described in Table 10-33.

Return to the Summary Table.

Destination Address Shadow Register

Figure 10-33. DST_ADDR_SHADOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
R/W-0h																															

Table 10-33. DST_ADDR_SHADOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Shadow Destination Address At the start of a transfer, the value in this register is loaded into the DST_ADDR_ACTIVE register and used as the value of the destination address. This register can be safely updated during a transfer. Reset type: SYSRSn

10.8.3.23 DST_BEG_ADDR_ACTIVE Register (Offset = 1Ch) [Reset = 00000000h]

DST_BEG_ADDR_ACTIVE is shown in Figure 10-34 and described in Table 10-34.

Return to the Summary Table.

Destination Begin Address Active Register

Figure 10-34. DST_BEG_ADDR_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEGADDR																															
R-0h																															

Table 10-34. DST_BEG_ADDR_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BEGADDR	R	0h	Active Destination Beginning Address If a transfer is ongoing, this register holds the current destination value for the source address. This address may be updated after wrapping. When a transfer starts, this register is loaded with the shadow address from the DST_BEG_ADDR_SHADOW register. Reset type: SYSRSn

10.8.3.24 DST_ADDR_ACTIVE Register (Offset = 1Eh) [Reset = 00000000h]

DST_ADDR_ACTIVE is shown in Figure 10-35 and described in Table 10-35.

Return to the Summary Table.

Destination Address Active Register

Figure 10-35. DST_ADDR_ACTIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															
R-0h																															

Table 10-35. DST_ADDR_ACTIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	Active Destination Address If a transfer is ongoing, this register holds the current value of the destination address. This address may change after a write, a burst, or wrapping. Reset type: SYSRSn

Chapter 11

Analog Subsystem

The analog subsystem module is described in this chapter.

11.1 Introduction

The analog modules on this device include the Analog-to-Digital Converter (ADC), Temperature Sensor, Lite Comparator Subsystem variant (CMPSS_LITE), and Programmable Gain Amplifier (PGA).

11.1.1 Features

The analog subsystem has the following features:

- Flexible voltage references
  - The ADC is referenced to VREFHI and VSSA pins
    - VREFHI pin voltage can be driven in externally or can be generated by an internal bandgap voltage reference. For the 32RHB and 32VFC package, VREFHI is internally tied to VDDA.
    - The internal voltage reference range can be selected to be 0V to 3.3V or 0V to 2.5V
  - The comparator DACs are referenced to VDDA and VSSA
- Flexible pin usage
  - Comparator subsystem inputs, PGA functions, and digital inputs (AIOs)/outputs (AGPIOs) are multiplexed with ADC inputs
  - Comparator DAC can optionally be buffered and brought out to a multiplexed ADC pin (mutually exclusive with use of CMPSS compare functions)
  - Internal connection to V_{REFLO} on ADC for offset self-calibration

11.1.2 Block Diagram

The following analog subsystem block diagrams show the connections between the different integrated analog modules to the device pins. These pins fall into two categories: analog module inputs/outputs and reference pins.

The analog pins are organized into analog groups around the CMPSS module. The pins which connect to CMPSS inputs can be used for the CMPSS without further action and without preventing use as an ADC input simultaneously.

Some analog pins support digital functionality through muxed AIOs and AGPIOs. AIOs support only digital input functionality while AGPIOs support full digital input and output functionality.

The following notes apply to all packages:

- Not all analog pins are available on all devices. See the device data sheet to determine which pins are available.
- See the device data sheet to determine the allowable voltage range for VREFHI and VREFLO.
- An external capacitor is required on the VREFHI pins. See the device data sheet for the specific value required.

Figure 11-2 shows how each analog group is structured. Section 11.4 lists the analog pins and internal connections.

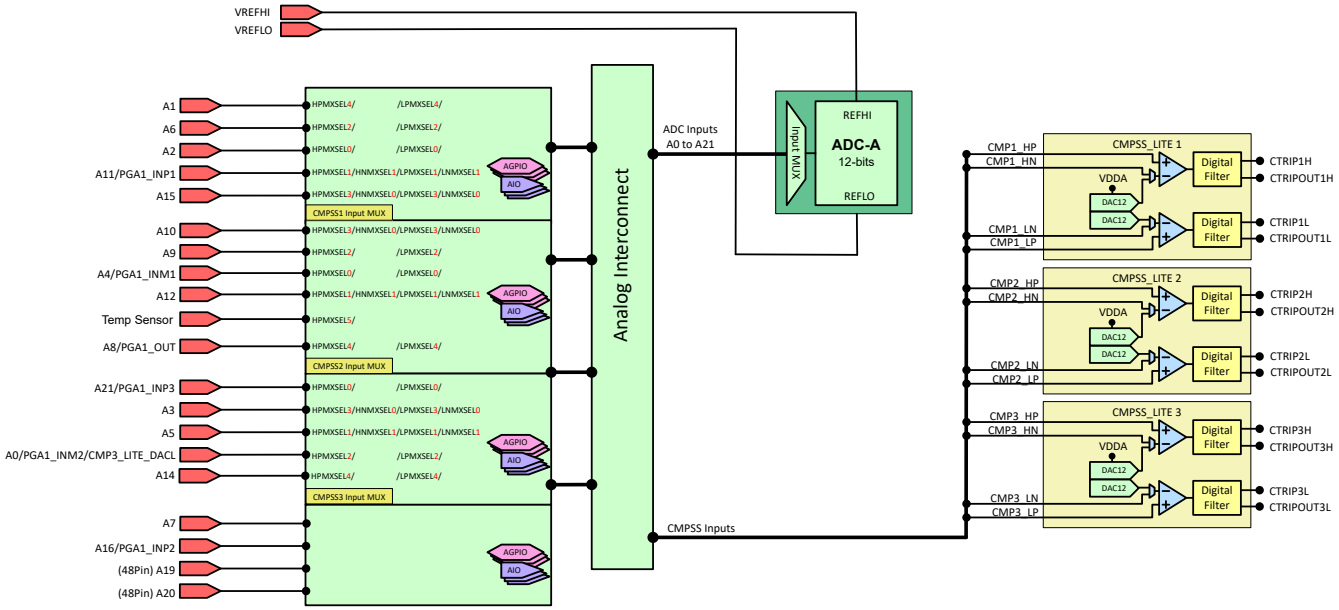
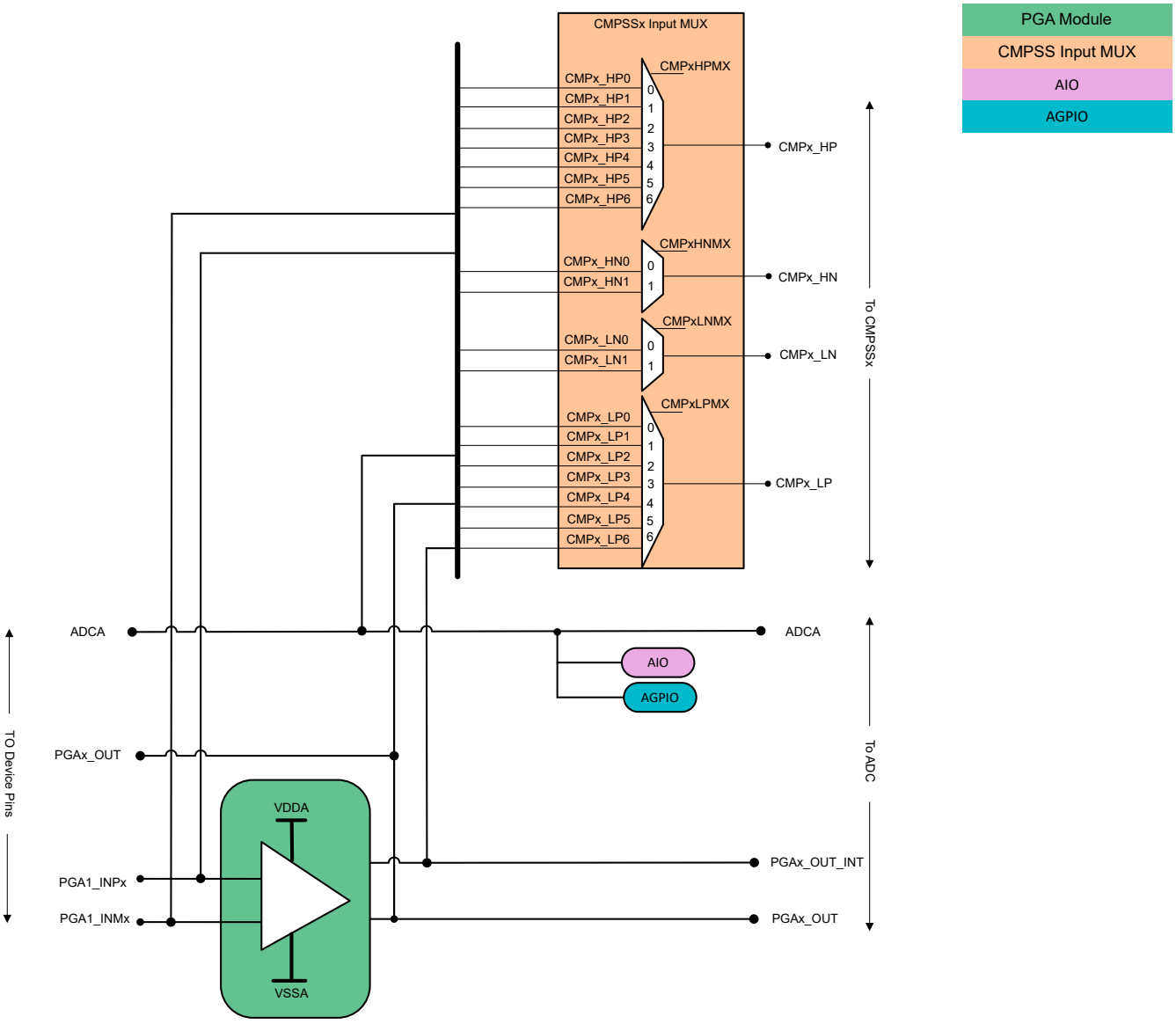


Figure 11-1. Analog Subsystem Block Diagram



Note: AIOs support digital input mode only.

Figure 11-2. Analog Group Connections

11.2 Digital Inputs on ADC Pins (AIOs)

Some GPIOs are multiplexed with analog pins and only have digital input functionality. These are also referred to as AIOs. Pins with only an AIO option on this port can only function in input mode. See the device data sheet for list of AIO signals. By default, these pins function as analog pins and the GPIOs are in a high-impedance state. The GPyAMSEL register is used to configure these pins for digital or analog operation.

Note

If digital signals with sharp edges (high dv/dt) are connected to the AIOs, cross-talk can occur with adjacent analog signals. Therefore, limit the edge rate of signals connected to AIOs if adjacent channels are being used for analog functions.

11.3 Digital Inputs and Outputs on ADC Pins (AGPIOs)

Some GPIOs are multiplexed with analog pins and have digital input and output functionality. These are also referred to as AGPIOs. Unlike AIOs, AGPIOs have full input and output capability. By default, the AGPIOs are not connected and must be configured. Table 11-1 shows how to configure the AGPIOs. To enable the analog functionality, set the register AGPIOCTRLx from analog subsystem. To enable the digital functionality, set the register GPxAMSEL from the *General-Purpose Input/Output (GPIO)* chapter.

Table 11-1. AGPIO Configuration

AGPIOCTRLx.GPIOy (Default = 0)	GPxAMSEL.GPIOy (Default = 1)	Pin Connected To:	
		ADC	GPIOy
0	0	-	Yes
0	**1**	**-** (1)	**-** (1)
1	0	-	Yes
1	1	Yes	-

(1) By default there are no signals connected to AGPIO pins. One of the other rows in the table must be chosen for pin functionality.

Note

If digital signals with sharp edges (high dv/dt) are connected to the AGPIOs, cross-talk can occur with adjacent analog signals. The user must therefore limit the edge rate of signals connected to AGPIOs, if adjacent channels are being used for analog functions.

11.4 Analog Pins and Internal Connections

Table 11-2. Analog Pins and Internal Connections

Pin Name	Pins/Package			ADC	DAC	PGA	Comparator Subsystem (Mux)				AIO Input/GPIO
	48 QFP	32 QFP	32 QFN				High Positive	High Negative	Low Positive	Low Negative	
VREFHI	12	-(4)	-(4)								
VREFLO	13	-(4)	-(4)								
Analog Group 1							**CMP1**				
A6	4(1)	2(1)	2(1)	A6			CMP1 (HPMXSEL=2)		CMP1 (LPMXSEL=2)		GPIO228(3)
A2	6	4	4	A2			CMP1 (HPMXSEL=0)		CMP1 (LPMXSEL=0)		GPIO224(3)
A15	7(1)	5(1)	5(1)	A15			CMP1 (HPMXSEL=3)	CMP1 (HNMXSEL=0)	CMP1 (LPMXSEL=3)	CMP1 (LNMXSEL=0)	AIO233
A11	8	6(1)	6(1)	A11		PGA_INP1	CMP1 (HPMXSEL=1)	CMP1 (HNMXSEL=1)	CMP1 (LPMXSEL=1)	CMP1 (LNMXSEL=1)	AIO237
A1	10	7(1)	7(1)	A1			CMP1 (HPMXSEL=4)		CMP1 (LPMXSEL=4)		AIO232
Analog Group 2							**CMP2**				
A10	21	13(1)	13(1)	A10			CMP2 (HPMXSEL=3)	CMP2 (HNMXSEL=0)	CMP2 (LPMXSEL=3)	CMP2 (LNMXSEL=0)	GPIO230(3)
A12	14	8(1)	8(1)	A12		PGA_INN3	CMP2 (HPMXSEL=1)	CMP2 (HNMXSEL=1)	CMP2 (LPMXSEL=1)	CMP2 (LNMXSEL=1)	AIO238
A8/PGA1_OUT	16	9	9	A8		PGA_OUT	CMP2 (HPMXSEL=4)		CMP2 (LPMXSEL=4)		AIO241
A4/PGA1_INM1	19	12	12	A4		PGA_INM1	CMP2 (HPMXSEL=0)		CMP2 (LPMXSEL=0)		AIO225
A9	20	13(1)	13(1)	A9			CMP2 (HPMXSEL=2)		CMP2 (LPMXSEL=2)		GPIO227(3)
Analog Group 3							**CMP3**				
A3	5	3	3	A3			CMP3 (HPMXSEL=3)	CMP3 (HNMXSEL=0)	CMP3 (LPMXSEL=3)	CMP3 (LNMXSEL=0)	GPIO242(3)
A14	7(1)	5(1)	5(1)	A14			CMP3 (HPMXSEL=4)		CMP3 (LPMXSEL=4)		AIO239
A5	9	6(1)	6(1)	A5			CMP3 (HPMXSEL=1)	CMP3 (HNMXSEL=1)	CMP3 (LPMXSEL=1)	CMP3 (LNMXSEL=1)	AIO244
A0/ CMP3_LITE_DACL/ PGA1_INM2	11	7(1)	7(1)	A0	CMP3_LITE_DACL	PGA1_INM2	CMP3 (HPMXSEL=2)		CMP3 (LPMXSEL=2)		AIO231
A21/PGA1_INP3	4(1)	2(1)	2(1)	A21			CMP3 (HPMXSEL=0)		CMP3 (LPMXSEL=0)		GPIO 226(3)

Table 11-2. Analog Pins and Internal Connections (continued)

Pin Name	Pins/Package			ADC	DAC	PGA	Comparator Subsystem (Mux)				AIO Input/GPIO
	48 QFP	32 QFP	32 QFN				High Positive	High Negative	Low Positive	Low Negative	
Other Analog											
A16/PGA1_INP2	2	32	32	A16		PGA1_INP2					GPIO28(3)
A19	23	-	-	A19							GPIO13(3)
A20	24	-	-	A20							GPIO12(3)
A7	15	8(1)	8(1)	A7							AIO245
TempSensor(2)	-	-	-	A22			CMP1 (HPMXSEL=5)				
PGA1_OUT_INT(2)	-	-	-	A25			CMP2 (HPMXSEL=6)		CMP2 (LPMXSEL=6)		

(1) Signal is bonded together with another signal as a single pin on this package.
(2) Internal connection only; does not come to a device pin.
(3) The GPIOs on these analog pins support full digital input and output functionality and are referred to as AGPIOs. By default, the AGPIOs are unconnected; that is, the analog and digital functions are both disabled. For configuration details, see the *Digital Inputs and Outputs on ADC Pins (AGPIOs)* section.
(4) On 32 RHB and 32 VFC package, VREFHI is internally connected to VDDA and VREFLO is internally connected to VSSA.

Note

The GPIOs on the analog pins support full digital input and output functionality and are referred to as AGPIOs. By default, the AGPIOs are unconnected; that is, the analog and digital functions are both disabled. For configuration details, see the *Digital Inputs and Outputs on ADC Pins (AGPIOs)* section.

Table 11-3. Analog Signal Descriptions

Signal Name	Description
AIOx	Digital input on ADC pin
AGPIOx	Digital input/output pin with ADC functionality
Ax	ADC A Input
CMPx_HNy	Comparator subsystem high comparator negative input
CMPx_HPy	Comparator subsystem high comparator positive input
CMPx_LNy	Comparator subsystem low comparator negative input
CMPx_LPy	Comparator subsystem low comparator positive input
CMP3_LITE_DACL	DAC output from the lower CMPSS3_LITE DAC (can be brought to an external pin)
PGAx_INPy	PGA module non-inverting pin
PGAx_INMy	PGA module inverting pin
PGAx_OUT	PGA module output
PGAx_OUT_INT	PGA module internal output connected to CMPSS and ADC modules
TempSensor	Internal temperature sensor

Table 11-4. Reference Summary

Module	Reference Option	Configured Where?	Register	Driverlib Function	Notes
ADC	Internal	Analog System	AnalogSubsysRegs. ANAREFCTL.bit. ANAREFxSEL	ADC_setVREF	Both options require use of the VREFHI pin.
	External	Analog System	1) AnalogSubsysRegs. ANAREFCTL.bit. ANAREFxSEL2) AnalogSubsysRegs. REFCONFIGA.bit. ANAREFSEL	ADC_setVREF	Both options require use of the VREFHI pin.
	3.3V or 2.5V Internal Reference Range	Analog System	AnalogSubsysRegs. ANAREFCTL.bit. ANAREFx2P5SEL	ADC_setVREF	Only applicable when using internal reference mode.
CMPSS DACs	VDDA	CMPSS Module	Not configurable		

11.5 ASBSYS Registers

This Section describes the ASBSYS Registers.

11.5.1 ASBSYS Base Address Table

Table 11-5. ASBSYS Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
AnalogSubsysRegs	ANALOG_SUBSYS_REGS	ANALOGSUBSYS_BASE	0x0005_D700	-	YES

11.5.2 ANALOG_SUBSYS_REGS Registers

Table 11-6 lists the memory-mapped registers for the ANALOG_SUBSYS_REGS registers. All register offset addresses not listed in Table 11-6 should be considered as reserved locations and the register contents should not be modified.

Table 11-6. ANALOG_SUBSYS_REGS Registers

Offset	Acronym	Register Name	Write Protection
2Ch	ADCOSDETECT	I2V Logic Control	EALLOW
3Ah	REFCONFIGA	Config register for analog reference A.	EALLOW
56h	INTERNALTESTCTL	INTERNALTEST Node Control Register	EALLOW
6Ah	CONFIGLOCK	Lock Register for all the config registers.	EALLOW
6Ch	TSNSCTL	Temperature Sensor Control Register	EALLOW
74h	ANAREFCTL	Analog Reference Control Register. This register is not configurable for 32QFN package	EALLOW
7Ch	VMONCTL	Voltage Monitor Control Register	EALLOW
8Eh	CMPHPMXSEL	Bits to select one of the many sources on CompHP inputs. Refer to Pimux diagram for details.	EALLOW
90h	CMPLPMXSEL	Bits to select one of the many sources on CompLP inputs. Refer to Pimux diagram for details.	EALLOW
92h	CMPHNMXSEL	Bits to select one of the many sources on CompHN inputs. Refer to Pimux diagram for details.	EALLOW
93h	CMPLNMXSEL	Bits to select one of the many sources on CompLN inputs. Refer to Pimux diagram for details.	EALLOW
94h	ADCDACLOOPBACK	Enabble loopback from DAC to ADCs	
97h	CMPSSCTL	CMPSS Control Register	EALLOW
9Ah	LOCK	Lock Register	EALLOW
120h	AGPIOCTRLA	AGPIO Control Register	EALLOW
12Eh	AGPIOCTRLH	AGPIO Control Register	EALLOW
140h	GPIOINENACTRL	GPIOINENACTRL Control Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 11-7 shows the codes that are used for access types in this section.

Table 11-7. ANALOG_SUBSYS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WOnce	W Once	Write Write once
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 11-7. ANALOG_SUBSYS_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

11.5.2.1 ADCOSDETECT Register (Offset = 2Ch) [Reset = 0000h]

ADCOSDETECT is shown in Figure 11-3 and described in Table 11-8.

Return to the Summary Table.

I2V Logic Control

Figure 11-3. ADCOSDETECT Register

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
DETECTCFG			OSDETECT_EN	RESERVED			
R/W-0h			R/W-0h	R/W-0h			

Table 11-8. ADCOSDETECT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R/W	0h	Reserved
7-5	DETECTCFG	R/W	0h	ADC Opens and Shorts Detect Configuration. This bit field defines the open/shorts detection circuit state. 0h Open/Shorts detection circuit is disabled. 1h Open/Shorts detection circuit is enabled at zero scale. 2h Open/Shorts detection circuit is enabled at full scale. 3h Open/Shorts detection circuit is enabled at (nominal) 5/12 scale. 4h Open/Shorts detection circuit is enabled at (nominal) 7/12 scale. 5h Open/Shorts detection circuit is enabled with a (nominal) 5K pulldown to VSSA. 6h Open/Shorts detection circuit is enabled with a (nominal) 5K pullup to VDDA. 7h Open/Shorts detection circuit is enabled with a (nominal) 7K pulldown to VSSA. Reset type: XRSn
4	OSDETECT_EN	R/W	0h	Set this bit to enable the OSDETECT logic Reset type: XRSn
3-0	RESERVED	R/W	0h	Reserved

11.5.2.2 REFCONFIGA Register (Offset = 3Ah) [Reset = 00000000h]

REFCONFIGA is shown in Figure 11-4 and described in Table 11-9.

Return to the Summary Table.

Config register for analog reference A.

Figure 11-4. REFCONFIGA Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED		RESERVED	RESERVED		
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED			ANAREFSEL	RESERVED	RESERVED		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
15	14	13	12	11	10	9	8
RESERVED	RESERVED			RESERVED			
R/W-0h	R/W-0h			R/W-0h			
7	6	5	4	3	2	1	0
RESERVED			RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-9. REFCONFIGA Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26-21	RESERVED	R/W	0h	Reserved
20	ANAREFSEL	R/W	0h	Analog reference mode select. This bit selects whether the VREFHI pin uses internal reference mode (the device drives a voltage onto the VREFHI pin) or external reference mode (the system is expected to drive a voltage into the VREFHI pin). 0 Internal reference mode 1 External reference mode Reset type: XRSn
19	RESERVED	R/W	0h	Reserved
18-15	RESERVED	R/W	0h	Reserved
14-12	RESERVED	R/W	0h	Reserved
11-5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

11.5.2.3 INTERNALTESTCTL Register (Offset = 56h) [Reset = 00000000h]

INTERNALTESTCTL is shown in Figure 11-5 and described in Table 11-10.

Return to the Summary Table.

INTERNALTEST Node Control Register

Figure 11-5. INTERNALTESTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED		TESTSEL					
R/W-0h		R/W-0h					

Table 11-10. INTERNALTESTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0/W	0h	Reserved
15-9	RESERVED	R-0	0h	Reserved
8-6	RESERVED	R/W	0h	Reserved

Table 11-10. INTERNALTESTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	TESTSEL	R/W	0h	Test Select. This bit field defines which internal node, if any, is selected to come out on the INTERNALTEST node connected to the ADC. Reset type: SYSRSn 0h (R/W) = No internal connection 1h (R/W) = VDDCORE 2h (R/W) = VDDA 3h (R/W) = VSSA 4h (R/W) = VREFLO 5h (R/W) = Reserved 6h (R/W) = Reserved 7h (R/W) = Reserved 8h (R/W) = Reserved 9h (R/W) = Reserved Ah (R/W) = Reserved Bh (R/W) = Reserved Ch (R/W) = Reserved 1Ch (R/W) = Reserved 1Dh (R/W) = Reserved 1Eh (R/W) = CMPSS1 VDDA sense on TESTANA0,VSSA sense on TESTANA1 1Fh (R/W) = ADCA VDDA sense on TESTANA0,VSSA sense on TESTANA1 20h (R/W) = COMP DAC BUFFER VDDA sense on TESTANA0,VSSA sense on TESTANA1 21h (R/W) = PGA1 VDDA sense on TESTANA0,VSSA sense on TESTANA1 22h (R/W) = Reserved 23h (R/W) = Reserved 24h (R/W) = Reserved 25h (R/W) = Reserved 28h (R/W) = Reserved 29h (R/W) = Reserved 2Ah (R/W) = Reserved 2Bh (R/W) = Reserved 2Ch (R/W) = Reserved 2Dh (R/W) = Reserved 30h (R/W) = Reserved 31h (R/W) = Reserved 3Fh (R/W) = Reserved

11.5.2.4 CONFIGLOCK Register (Offset = 6Ah) [Reset = 00000000h]

CONFIGLOCK is shown in Figure 11-6 and described in Table 11-11.

Return to the Summary Table.

Lock Register for all the config registers.

Figure 11-6. CONFIGLOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	GPIOINENACTRL	RESERVED	RESERVED	AGPIOCTRL	RESERVED	RESERVED	RESERVED
R-0-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 11-11. CONFIGLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6	GPIOINENACTRL	R/WSonce	0h	Locks all GPIOINENACTRL Register. Setting this bit will disable any future writes to this reigster. This bit can only be cleared by a reset. Reset type: SYSRSn
5	RESERVED	R/WSonce	0h	Reserved
4	RESERVED	R/WSonce	0h	Reserved
3	AGPIOCTRL	R/WSonce	0h	Locks all AGPIOCTRL Register. Setting this bit will disable any future writes to this reigster. This bit can only be cleared by a reset. Reset type: SYSRSn
2	RESERVED	R/WSonce	0h	Reserved
1	RESERVED	R/WSonce	0h	Reserved
0	RESERVED	R/WSonce	0h	Reserved

11.5.2.5 TSNSCTL Register (Offset = 6Ch) [Reset = 0000h]

TSNSCTL is shown in Figure 11-7 and described in Table 11-12.

Return to the Summary Table.

Temperature Sensor Control Register

Figure 11-7. TSNSCTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							ENABLE
R-0-0h							R/W-0h

Table 11-12. TSNSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-1	RESERVED	R-0	0h	Reserved
0	ENABLE	R/W	0h	Temperature Sensor Enable. This bit enables the temperature sensor output to the ADC. 0 Disabled 1 Enabled Reset type: SYSRSn

11.5.2.6 ANAREFCTL Register (Offset = 74h) [Reset = 000Fh]

ANAREFCTL is shown in Figure 11-8 and described in Table 11-13.

Return to the Summary Table.

Analog Reference Control Register. This register is not configurable for 32QFN package

Figure 11-8. ANAREFCTL Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED				RESERVED	RESERVED	ANAREF2P5SEL
R/W-0h	R-0-0h				R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED					RESERVED	RESERVED	RESERVED
R-0-1h					R/W-1h	R/W-1h	R/W-1h

Table 11-13. ANAREFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R/W	0h	Reserved
14-11	RESERVED	R-0	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	ANAREF2P5SEL	R/W	0h	Analog reference A 2.5V source select. In internal reference mode, this bit selects which voltage the internal reference buffer drives onto the VREFHI pin. The buffer can drive either 1.65V onto the pin, resulting in a reference range of 0 to 3.3V, or the buffer can drive 2.5V onto the pin, resulting in a reference range of 0 to 2.5V. If switching between these two modes, the user must allow adequate time for the external capacitor to charge to the new voltage before using the ADC or buffered DAC. 0 Internal 1.65V reference mode (3.3V reference range) 1 Internal 2.5V reference mode (2.5V reference range) Reset type: XRSn
7-3	RESERVED	R-0	1h	Reserved
2	RESERVED	R/W	1h	Reserved
1	RESERVED	R/W	1h	Reserved
0	RESERVED	R/W	1h	Reserved

11.5.2.7 VMONCTL Register (Offset = 7Ch) [Reset = 0000h]

VMONCTL is shown in Figure 11-9 and described in Table 11-14.

Return to the Summary Table.

Voltage Monitor Control Register

Figure 11-9. VMONCTL Register

15	14	13	12	11	10	9	8
RESERVED							BORLVMONDIS
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED						RESERVED	RESERVED
R-0-0h						R/W-0h	R/W-0h

Table 11-14. VMONCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R-0	0h	Reserved
8	BORLVMONDIS	R/W	0h	BORL disable on VDDIO. 0 BORL is enabled on VDDIO, i.e BOR circuit will be triggered if VDDIO goes lower than the lower BOR threshold of VDDIO. 1 BORL is disabled on VDDIO, i.e BOR circuit will not be triggered if VDDIO goes lower than the lower BOR threshold of VDDIO. Reset type: SYSRSn
7-2	RESERVED	R-0	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

11.5.2.8 CMPHPMXSEL Register (Offset = 8Eh) [Reset = 00000000h]

CMPHPMXSEL is shown in Figure 11-10 and described in Table 11-15.

Return to the Summary Table.

Bits to select one of the many sources on CompHP inputs. Refer to Pimux diagram for details.

Figure 11-10. CMPHPMXSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED			RESERVED			RESERVED	
R-0-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	RESERVED			RESERVED			CMP3HPMXSEL
R-0-0h	R/W-0h			R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
CMP3HPMXSEL		CMP2HPMXSEL			CMP1HPMXSEL		
R/W-0h		R/W-0h			R/W-0h		

Table 11-15. CMPHPMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	Reserved
21-19	RESERVED	R/W	0h	Reserved
18-16	RESERVED	R/W	0h	Reserved
15	RESERVED	R-0	0h	Reserved
14-12	RESERVED	R/W	0h	Reserved
11-9	RESERVED	R/W	0h	Reserved
8-6	CMP3HPMXSEL	R/W	0h	CMP3HPMXSEL bits, Refer to the Analog Subsystem chapter Note: Only values 0 to 4 and '6' are valid, rest are reserved Reset type: XRSn
5-3	CMP2HPMXSEL	R/W	0h	CMP2HPMXSEL bits, Refer to the Analog Subsystem chapter Note: Only values 0 to 6 are valid, rest are reserved Reset type: XRSn
2-0	CMP1HPMXSEL	R/W	0h	CMP1HPMXSEL bits, Refer to the Analog Subsystem chapter Note: Only values 0 to 4 and '6' are valid, rest are reserved Reset type: XRSn

11.5.2.9 CMPLPMXSEL Register (Offset = 90h) [Reset = 00000000h]

CMPLPMXSEL is shown in Figure 11-11 and described in Table 11-16.

Return to the Summary Table.

Bits to select one of the many sources on CompLP inputs. Refer to Pimux diagram for details.

Figure 11-11. CMPLPMXSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED			RESERVED			RESERVED	
R-0-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	RESERVED			RESERVED			CMP3LPMXSEL
R-0-0h	R/W-0h			R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
CMP3LPMXSEL		CMP2LPMXSEL			CMP1LPMXSEL		
R/W-0h		R/W-0h			R/W-0h		

Table 11-16. CMPLPMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R-0	0h	Reserved
21-19	RESERVED	R/W	0h	Reserved
18-16	RESERVED	R/W	0h	Reserved
15	RESERVED	R-0	0h	Reserved
14-12	RESERVED	R/W	0h	Reserved
11-9	RESERVED	R/W	0h	Reserved
8-6	CMP3LPMXSEL	R/W	0h	CMP3LPMXSEL bits, Refer to the Analog Subsystem chapter Note: Only values 0 to 4 and '6' are valid, rest are reserved Reset type: XRSn
5-3	CMP2LPMXSEL	R/W	0h	CMP2LPMXSEL bits, Refer to the Analog Subsystem chapter Note: Only values 0 to 6 are valid, rest are reserved Reset type: XRSn
2-0	CMP1LPMXSEL	R/W	0h	CMP1LPMXSEL bits, Refer to the Analog Subsystem chapter Note: Only values 0 to 4 and '6' are valid, rest are reserved Reset type: XRSn

11.5.2.10 CMPHNMXSEL Register (Offset = 92h) [Reset = 0000h]

CMPHNMXSEL is shown in Figure 11-12 and described in Table 11-17.

Return to the Summary Table.

Bits to select one of the many sources on CompHN inputs. Refer to Pimux diagram for details.

Figure 11-12. CMPHNMXSEL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CMP3HNMXSEL	CMP2HNMXSEL	CMP1HNMXSEL
R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-17. CMPHNMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R-0	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	CMP3HNMXSEL	R/W	0h	CMP3HNMXSEL bits, Refer to the Analog Subsystem chapter Reset type: XRSn
1	CMP2HNMXSEL	R/W	0h	CMP2HNMXSEL bits, Refer to the Analog Subsystem chapter Reset type: XRSn
0	CMP1HNMXSEL	R/W	0h	CMP1HNMXSEL bits, Refer to the Analog Subsystem chapter Reset type: XRSn

11.5.2.11 CMPLNMXSEL Register (Offset = 93h) [Reset = 0000h]

CMPLNMXSEL is shown in Figure 11-13 and described in Table 11-18.

Return to the Summary Table.

Bits to select one of the many sources on CompLN inputs. Refer to Pimux diagram for details.

Figure 11-13. CMPLNMXSEL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CMP3LNMXSEL	CMP2LNMXSEL	CMP1LNMXSEL
R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-18. CMPLNMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R-0	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	CMP3LNMXSEL	R/W	0h	CMP3LNMXSEL bits, Refer to the Analog Subsystem chapter Reset type: XRSn
1	CMP2LNMXSEL	R/W	0h	CMP2LNMXSEL bits, Refer to the Analog Subsystem chapter Reset type: XRSn
0	CMP1LNMXSEL	R/W	0h	CMP1LNMXSEL bits, Refer to the Analog Subsystem chapter Reset type: XRSn

11.5.2.12 ADCDACLOOPBACK Register (Offset = 94h) [Reset = 00000000h]

ADCDACLOOPBACK is shown in Figure 11-14 and described in Table 11-19.

Return to the Summary Table.

Enabble loopback from DAC to ADCs

Figure 11-14. ADCDACLOOPBACK Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					RESERVED	RESERVED	ENLB2ADCA
R-0-0h					R/W-0h	R/W-0h	R/W-0h

Table 11-19. ADCDACLOOPBACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write Key. Writes to this register must include the value 0xA5A5 in the KEY bit field to take effect. Otherwise the register will remain as it was prior to the write attempt. Reads will return a 0. Reset type: XRSn
15-3	RESERVED	R-0	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	ENLB2ADCA	R/W	0h	1 Loops back COMPDAC output to ADCA. 0 Loop back is broken. Note: Setting this bit to 1, will override the CHSEL specification for the ADC. ADC would sample COMPDAC output irrespective of the value of CHSEL. Reset type: XRSn

11.5.2.13 CMPSSCTL Register (Offset = 97h) [Reset = 0000h]

CMPSSCTL is shown in Figure 11-15 and described in Table 11-20.

Return to the Summary Table.

CMPSS Control Register

Figure 11-15. CMPSSCTL Register

15	14	13	12	11	10	9	8
CMPSSCTLEN	RESERVED						
R/W-0h	R-0-0h						
7	6	5	4	3	2	1	0
RESERVED							CMP3LDACOUTEN
R-0-0h							R/W-0h

Table 11-20. CMPSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CMPSSCTLEN	R/W	0h	0 - Rest of the configurations in this register are disabled 1 - Rest of the configuration in this register are enabled This bit is added for safety purpose. The configurations in this register are donot care if this bit is '0' Reset type: SYSRSn
14-1	RESERVED	R-0	0h	Reserved
0	CMP3LDACOUTEN	R/W	0h	0 - CMPSS3.COMPL is enabled and associated DAC will act as reference for comparator. 1 - CMPSS3.COMPL is disabled. Associated DAC will act as a general purpose DAC Reset type: SYSRSn

11.5.2.14 LOCK Register (Offset = 9Ah) [Reset = 00000000h]

LOCK is shown in Figure 11-16 and described in Table 11-21.

Return to the Summary Table.

Lock Register

Figure 11-16. LOCK Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED					CMPSSCTL	RESERVED	CMPLNMXSEL
R-0-0h					R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
CMPHNMXSEL	CMPLPMXSEL	CMPHPMXSEL	RESERVED	RESERVED	VMONCTL	ANAREFCTL	TSNSCTL
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 11-21. LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	Reserved
10	CMPSSCTL	R/WSonce	0h	CMPSSCTL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
9	RESERVED	R/WSonce	0h	Reserved
8	CMPLNMXSEL	R/WSonce	0h	CMPLNMXSEL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
7	CMPHNMXSEL	R/WSonce	0h	CMPHNMXSEL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
6	CMPLPMXSEL	R/WSonce	0h	CMPLPMXSEL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
5	CMPHPMXSEL	R/WSonce	0h	CMPHPMXSEL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
4	RESERVED	R/WSonce	0h	Reserved
3	RESERVED	R/WSonce	0h	Reserved
2	VMONCTL	R/WSonce	0h	VMONCTL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn

Table 11-21. LOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	ANAREFCTL	R/WSonce	0h	ANAREFCTL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn
0	TSNSCTL	R/WSonce	0h	TSNSCTL Register Lock. Setting this bit will disable any future write to the respective register. This bit can only be cleared by a reset. Reset type: SYSRSn

11.5.2.15 AGPIOCTRLA Register (Offset = 120h) [Reset = 00000000h]

AGPIOCTRLA is shown in Figure 11-17 and described in Table 11-22.

Return to the Summary Table.

AGPIO Control Register

Figure 11-17. AGPIOCTRLA Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	GPIO28	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	GPIO13	GPIO12	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-22. AGPIOCTRLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	GPIO28	R/W	0h	One time configuration for GPIO28 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	GPIO13	R/W	0h	One time configuration for GPIO13 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn

Table 11-22. AGPIOCTRLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	GPIO12	R/W	0h	One time configuration for GPIO12 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

11.5.2.16 AGPIOCTRLH Register (Offset = 12Eh) [Reset = 00000000h]

AGPIOCTRLH is shown in Figure 11-18 and described in Table 11-23.

Return to the Summary Table.

AGPIO Control Register

Figure 11-18. AGPIOCTRLH Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED	RESERVED	GPIO243	GPIO242	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	GPIO230	RESERVED	GPIO228	GPIO227	GPIO226	RESERVED	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 11-23. AGPIOCTRLH Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29	RESERVED	R/W	0h	Reserved
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	RESERVED	R/W	0h	Reserved
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	GPIO243	R/W	0h	One time configuration for GPIO243 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
18	GPIO242	R/W	0h	One time configuration for GPIO242 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved

Table 11-23. AGPIOCTRLH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	GPIO230	R/W	0h	One time configuration for GPIO230 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
5	RESERVED	R/W	0h	Reserved
4	GPIO228	R/W	0h	One time configuration for GPIO228 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
3	GPIO227	R/W	0h	One time configuration for GPIO227 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
2	GPIO226	R/W	0h	One time configuration for GPIO226 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn
1	RESERVED	R/W	0h	Reserved
0	GPIO224	R/W	0h	One time configuration for GPIO224 to decide whether AGPIO functionality is enabled 0 - AGPIO functionality is disabled 1 - AGPIO functionality is enabled Reset type: XRSn

11.5.2.17 GPIOINENACTRL Register (Offset = 140h) [Reset = 00000007h]

GPIOINENACTRL is shown in Figure 11-19 and described in Table 11-24.

Return to the Summary Table.

GPIOINENACTRL Control Register

Figure 11-19. GPIOINENACTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					GPIO45	GPIO44	GPIO43
R-0-0h					R/W-1h	R/W-1h	R/W-1h

Table 11-24. GPIOINENACTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	GPIO45	R/W	1h	One time configuration for GPIO45 to decide whether Input buffer (INENA control) is enabled or disabled 0 - Input buffer is disabled 1 - Input buffer is enabled Reset type: XRSn
1	GPIO44	R/W	1h	One time configuration for GPIO44 to decide whether Input buffer (INENA control) is enabled or disabled 0 - Input buffer is disabled 1 - Input buffer is enabled Reset type: XRSn
0	GPIO43	R/W	1h	One time configuration for GPIO43 to decide whether Input buffer (INENA control) is enabled or disabled 0 - Input buffer is disabled 1 - Input buffer is enabled Reset type: XRSn

Chapter 12

Analog-to-Digital Converter (ADC)

The analog-to-digital converter (ADC) module described in this chapter is a Type 7 ADC. See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with modules of the same type, to determine the differences between the types, and for a list of device-specific differences within a type.

12.1 Introduction

The ADC module is a 12-bit successive approximation (SAR) style ADC . The ADC is composed of a core and a wrapper. The core is composed of the analog circuits which include the channel select MUX, the sample-and-hold (S/H) circuit, the successive approximation circuits, voltage reference circuits, and other analog support circuits. The wrapper is composed of the digital circuits that configure and control the ADC. These circuits include the logic for programmable conversions, result registers, interfaces to analog circuits, interfaces to the peripheral buses, post-processing circuits, and interfaces to other on-chip modules.

Each ADC module consists of a single sample-and-hold (S/H) circuit. The ADC module is designed to be duplicated multiple times on the same chip, allowing simultaneous sampling or independent operation of multiple ADCs. The ADC wrapper is start-of-conversion (SOC) based (see Section 12.3).

12.1.1 Features

The ADC supports the following features:

- 12-bit resolution
- Ratiometric external reference set by VREFHI and VREFLO pins
- Selectable internal reference of 2.5V or 3.3V
- Single-ended (SE) signal conversions
- Input multiplexer with up to 17 (32-pin package) or 19 (48-pin package) channels
- 16 configurable SOCs
- 16 individually addressable result registers
- One trigger repeater module, enabling customizable hardware oversampling modes with little or no CPU overhead
- Multiple trigger sources:
  - S/W - software immediate start
  - All MCPWMs- ADCSOC A, B, C or D
  - GPIO XINT2
  - CPU Timers 0/1/2
  - ADCINT1/2
- Three flexible PIE interrupts and DMA triggers
- Configurable interrupt placement
- Three post-processing blocks, each with:
  - Saturating offset calibration
  - Error from set-point calculation
  - High, low, zero-crossing and in-limit compare, with interrupt and MCPWM trip capability
  - Aggregation functions: sum and average (binary shift) (on PPB1 only)

12.1.2 ADC Related Collateral

Foundational Materials

- *ADC Input Circuit Evaluation for C2000 MCUs (PSPICE for TI) Application Report*
- *ADC Input Circuit Evaluation for C2000 MCUs (TINA-TI) Application Report*
- C28x Academy - ADC
- PSpice for TI design and simulation tool
- *Real-Time Control Reference Guide*
  - Refer to the ADC section
- TI Precision Labs - ADCs
- TI Precision Labs: Driving the reference input on a SAR ADC (Video)
- TI Precision Labs: Introduction to analog-to-digital converters (ADCs) (Video)
- TI Precision Labs: SAR ADC input driver design (Video)
- TI e2e: Connecting VDDA to VREFHI
- TI e2e: Topologies for ADC Input Protection
- TI e2e: Why does the ADC Input Voltage drop with sampling?

  - Sampling a high impedance voltage divider with ADC
- *Understanding Data Converters Application Report*
- *VREFHI Input Driver Design for C2000 MCUs Application Report*

Getting Started Materials

- ADC-PWM Synchronization Using ADC Interrupt
  - NOTE: This is a non-TI (third party) site.
- Analog-to-Digital Converter (ADC) Training for C2000 MCUs (Video)
- *C2000 ADC (Type-3) Performance Versus ACQPS Application Report*
- *Hardware Design Guide for F2800x C2000 Real-Time MCU Series*

Expert Materials

- *A potential firmware mistake may lead to control instability*
- *ADC Oversampling Application Report*
- *An Overview of Designing Analog Interface With TM320F28xx/28xxx DSCs Application Report*
- Analog Engineer's Calculator
- *Analog Engineer's Pocket Reference*
- *Charge-Sharing Driving Circuits for C2000 ADCs (using PSPICE-FOR-TI) Application Report*
- *Charge-Sharing Driving Circuits for C2000 ADCs (using TINA-TI) Application Report*
- Debugging an integrated ADC in a microcontroller using an oscilloscope
- Hardware oversampling using C2000 ADC (Video)
- *Methods for Mitigating ADC Memory Cross-Talk Application Report*
- TI Precision Labs: ADC AC specifications (Video)
- TI Precision Labs: ADC Error sources (Video)
- TI Precision Labs: ADC Noise (Video)
- TI Precision Labs: Analog-to-digital converter (ADC) drive topologies (Video)
- TI Precision Labs: Electrical overstress on data converters (Video)
- TI Precision Labs: High-speed ADC fundamentals (Video)
- TI Precision Labs: SAR & Delta-Sigma: Understanding the Difference (Video)
- TI e2e: ADC Bandwidth Clarification
- TI e2e: ADC Resolution with Oversampling
- TI e2e: ADC configuration for interleaved mode
- TI e2e: Simultaneous Sampling with Single ADC

12.1.3 Block Diagram

Figure 12-1 shows the block diagram for the ADC core and ADC wrapper.

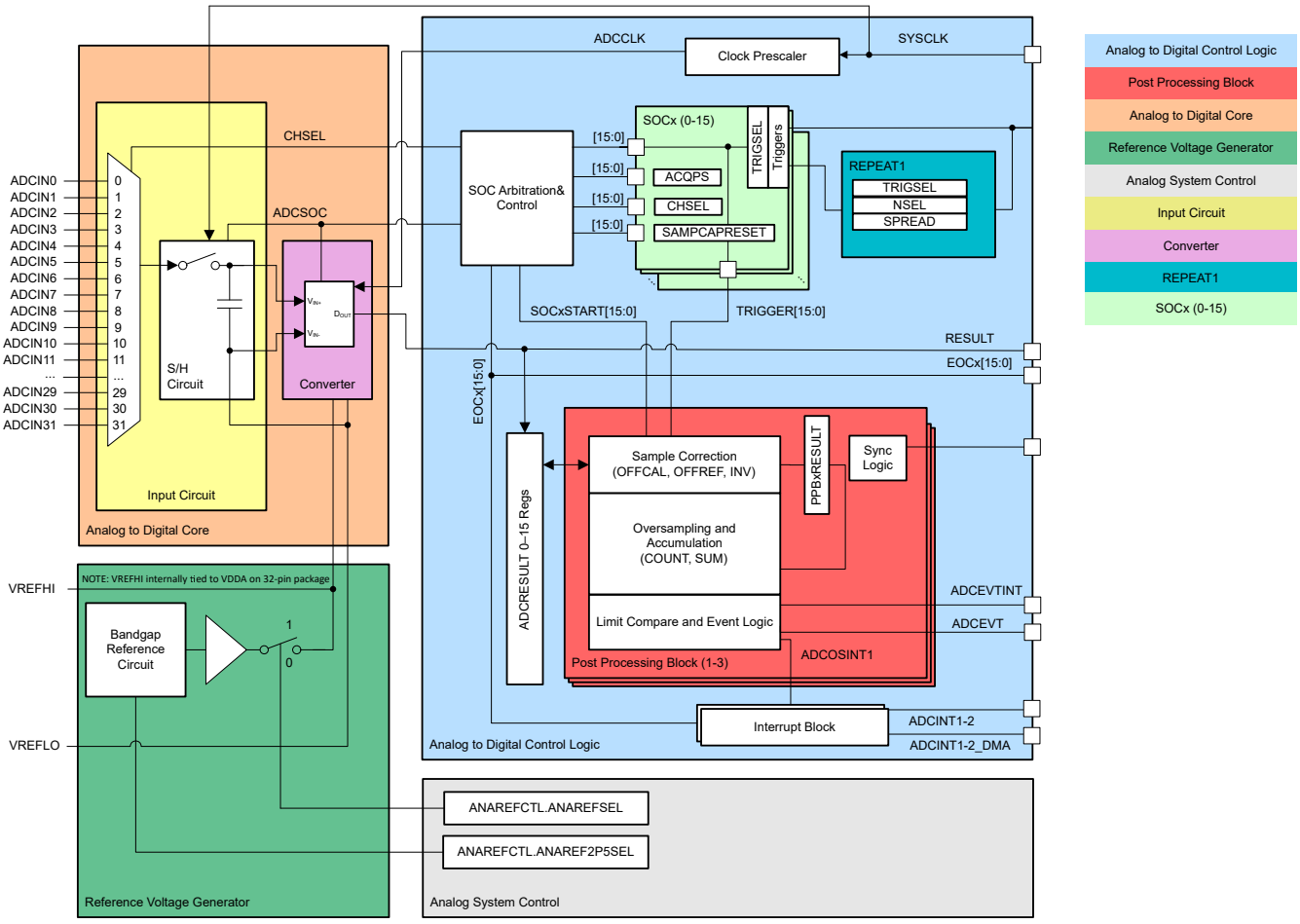


Figure 12-1. ADC Module Block Diagram

Note

- The ADC block diagram reflects the number of ADC channels internally configurable on the device. The actual number of available external ADC inputs varies depending on device part number and package type.
- To enable the external reference mode, the bit ANAREFSEL in register REFCONFIGA from AnalogSubsysRegs must be set to 1 (AnalogSubsysRegs.REFCONFIGA.bit.ANAREFSEL = 1).
- The Oversampling and Accumulation block shown in the post-processing block (PPB) is available only in one of the PPB instances.

12.2 ADC Configurability

Some ADC configurations are individually controlled by the SOCs, while others are globally controlled per ADC module. Table 12-1 summarizes the basic ADC options and the level of configurability. The subsequent sections discuss these configurations.

Table 12-1. ADC Options and Configuration Levels

Options	Configurability
Clock	Per module
Resolution	Not configurable (12-bit only)
Signal Mode	Not configurable (single-ended only)
Reference Voltage Source	Either external or internal
Trigger Source	Per SOC
Converted Channel	Per SOC
Acquisition Window Duration	Per SOC
EOC Location	Per module
Sample capacitor reset	Per SOC

12.2.1 ADC Clock Configuration

The base ADC clock is provided directly by the system clock. SYSCLK is used to generate the ADC acquisition window. The register ADCCTL2 has a PRESCALE field that determines the ADCCLK. ADCCLK is used to clock the converter, and is only active during the conversion phase. At all other times, including during the sample-and-hold window, the ADCCLK signal is gated off.

The core requires approximately 12 ADCCLK cycles to process a voltage into a conversion result. The user must determine the required duration of the acquisition window, see Section 12.12.1.

Note

To determine an appropriate value for ADCCTL2.PRESCALE, see the device data sheet for the maximum allowable ADCCLK frequencies.

12.2.2 Resolution

The resolution of the ADC determines how finely the analog range is quantized into digital values. The ADC module supports a fixed resolution of 12 bits.

12.2.3 Voltage Reference

12.2.3.1 External Reference Mode

The ADC module share VREFHI and VREFLO inputs. In external reference mode, these pins are used as a ratiometric reference to determine the ADC conversion input range.

See Section 12.12.4 for information on how to supply the reference voltage.

Note

- On devices with no external VREFHI and VREFLO pin, VREFHI is internally tied to VDDA and VREFLO is internally connected to the device analog ground, VSSA. Internal reference mode is not available for packages lacking VREFHI and VREFLO pins. See the device data sheet for packages with VREFHI and VREFLO pins available.
- To enable the external reference mode, the bit ANAREFSEL in register REFCONFIGA from AnalogSubsysRegs must be set to 1 (AnalogSubsysRegs.REFCONFIGA.bit.ANAREFSEL = 1).
- See the device data sheet to determine the allowable voltage range for VREFHI and VREFLO.
- The external reference mode requires an external capacitor and resistor on the VREFHI pin. See the device data sheet for the specific values required.

12.2.3.2 Internal Reference Mode

In internal reference mode, the device drives a voltage out onto the VREFHI pin. The VREFHI and VREFLO pins then set the ADC conversion range. Internal reference mode is not available for packages without VREFHI and VREFLO pins. See the device data sheet for packages with VREFHI and VREFLO pins available.

The internal reference voltage can be configured to be either 2.5V or 1.65V. When the 1.65V internal reference voltage is selected, the ADC input signal is internally divided by 2 before conversion, which effectively makes the ADC conversion range from VREFLO to 3.3V.

Note

The internal reference mode also requires an external capacitor and resistor on the VREFHI pin. See the device data sheet for the specific values required.

12.2.3.3 Selecting Reference Mode

The voltage reference mode must be configured by using either the ADC_setVREF() or the SetVREF() functions, depending on the header files used, provided in the C2000Ware Device SDK. Using either of these functions makes sure that the correct trim is loaded in the ADC trim registers. This function must be called at least once after a device reset. Do not configure the voltage reference mode by directly writing to the ANAREFCTL register.

12.2.4 Signal Mode

The ADC supports single-ended signaling.

In single-ended mode, the input voltage to the converter is sampled through a single pin (ADCINx), referenced to VREFLO.

12.2.4.1 Expected Conversion Results

Based on a given analog input voltage, the expected digital conversion is given in Table 12-2. Fractional values are truncated.

Table 12-2. Analog to 12-bit Digital Formulas

Analog Input	Digital Result
when ADCINy ≤ VREFLO	ADCRESULTx = 0
when VREFLO < ADCINy < VREFHI	$\text{ADCRESULTx} = 4096 \left(\frac{\text{ADCINy - VREFLO}}{\text{VREFHI - VREFLO}} \right)$
when ADCINy ≥ VREFHI	ADCRESULTx = 4095

12.2.4.2 Interpreting Conversion Results

Based on a given ADC conversion result, the corresponding analog input is given in Table 12-3. This corresponds to the center of the possible range of analog voltages that can produce this conversion result.

Table 12-3. 12-Bit Digital-to-Analog Formulas

Digital Value	Analog Equivalent	
when ADCRESULTy = 0	$\text{ADCINx} \leq \text{VREFLO}$	(2)
when 0 < ADCRESULTy < 4095	$\text{ADCINx} = (\text{VREFHI} - \text{VREFLO})\left(\frac{\text{ADCRESULTy}}{4096}\right) + \text{VREFLO}$	(3)
when ADCRESULTy = 4095	$\text{ADCINx} \geq \text{VREFHI}$	(4)

12.3 SOC Principle of Operation

The ADC triggering and conversion sequencing is accomplished through configurable start-of-conversions (SOCs). Each SOC is a configuration set defining the single conversion of a single channel. In that set, there are three configurations: the trigger source that starts the conversion, the channel to convert, and the acquisition (sample) window duration. Upon receiving the trigger configured for a SOC, the wrapper makes sure that the specified channel is captured using the specified acquisition window duration.

Multiple SOCs can be configured for the same trigger, channel, and acquisition window as desired. Configuring multiple SOCs to use the same trigger allows the trigger to generate a sequence of conversions. Configuring multiple SOCs to use the same trigger and channel allows for oversampling. Oversampling can also be achieved using a single trigger source by configuring the trigger repeater module. See Section 12.3.2.1 for more information.

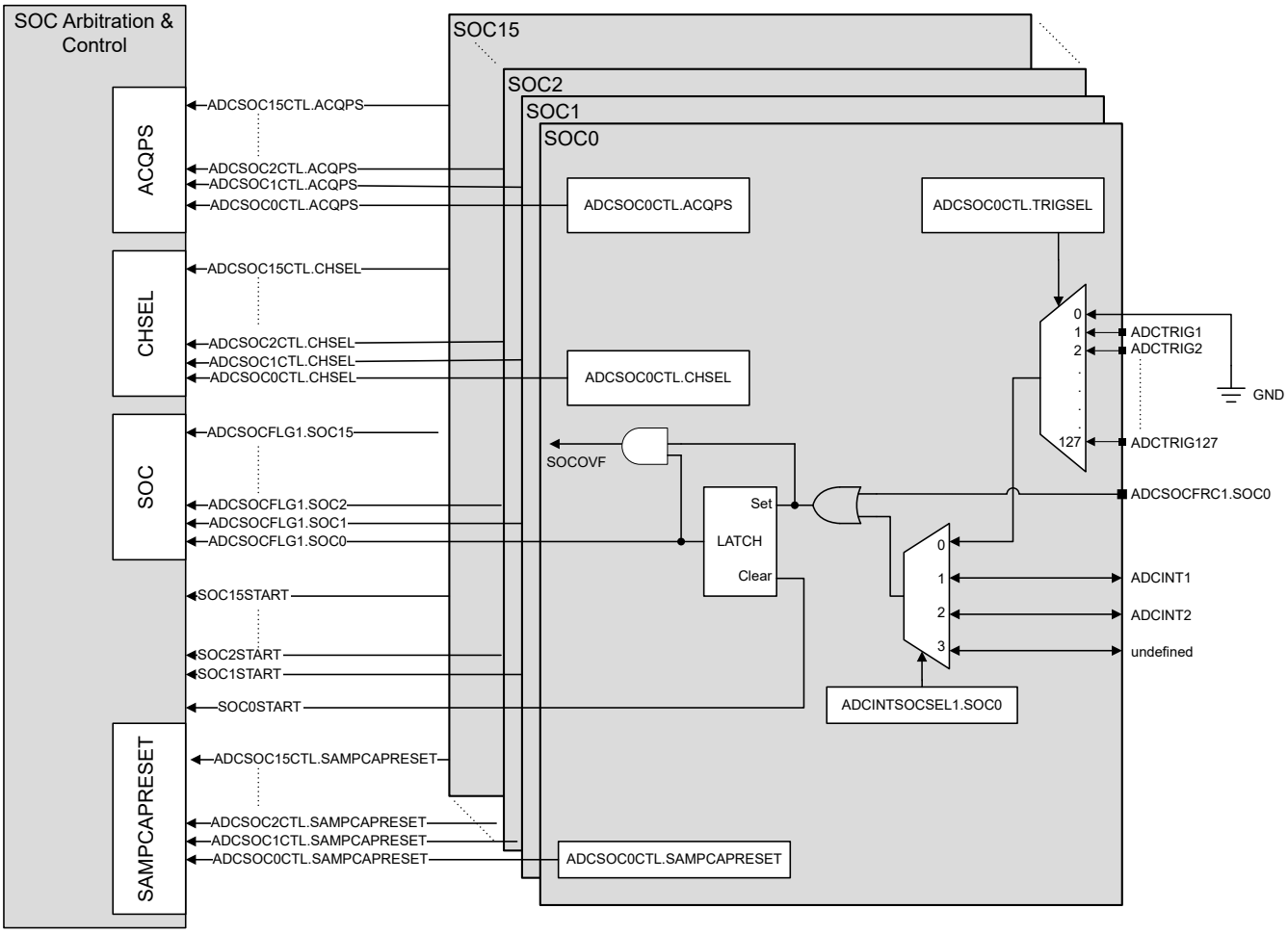


Figure 12-2. SOC Block Diagram

12.3.1 SOC Configuration

Each SOC has a configuration register, ADCSOCxCTL. Within this register, SOCx can be configured for a specific trigger source, channel to convert, and acquisition (sample) window duration.

12.3.2 Trigger Operation

Each SOC can be configured to start on one of many input triggers. The primary trigger select for SOCx is in the ADCSOCxCTL.TRIGSEL register, which can select between:

- Disabled (Software only)
- CPU Timers 0/1/2
- GPIO: INPUTXBAR.OUT[5]
- ADCSOCA, ADCSOCB, ADCSOCC, or ADCSOCD from each MCPWM module
- The trigger repeater block. This can be used to achieve oversampling and re-trigger spread

In addition, each SOC can also be triggered when the ADCINT1 flag or ADCINT2 flag is set. This is achieved by configuring the ADCINTSOCSEL1 register (for SOC0 to SOC7) or the ADCINTSOCSEL2 register (for SOC8 to SOC15). This can be useful for creating continuous conversions.

Table 12-4. ADC SOC Trigger Selection

ADCSOCCTL.TRIGSEL	Signal
0	ADC_SOFTWARE_TRIGGER
1	CPU1_TINT0
2	CPU1_TINT1
3	CPU1_TINT2
4	INPUTXBAR5
5	PWM1_SOCA
6	PWM1_SOCB
7	PWM1_SOCC
8	PWM1_SOCD
9	
10	
11	
12	
13	PWM3_SOCA
14	PWM3_SOCB
15	
16	
17	
18	
19	
20	
21	REP1TRIG
22-31	Reserved

12.3.2.1 Trigger Repeaters

The ADC contains one trigger repeater module. This module can select any of the regular ADC triggers that are selectable by the ADCSOCxCTL.TRIGGER register, and generate a number of repeat pulses as configured in the REP1N.NSEL register. Figure 12-3 shows a functional block diagram of the ADC trigger repeater module.

The repeater module can apply two types of trigger modifications:

- Oversampling mode (Up to 8 samples)
- Re-trigger spread

Each of these trigger modification features is explained in detail in the following sections.

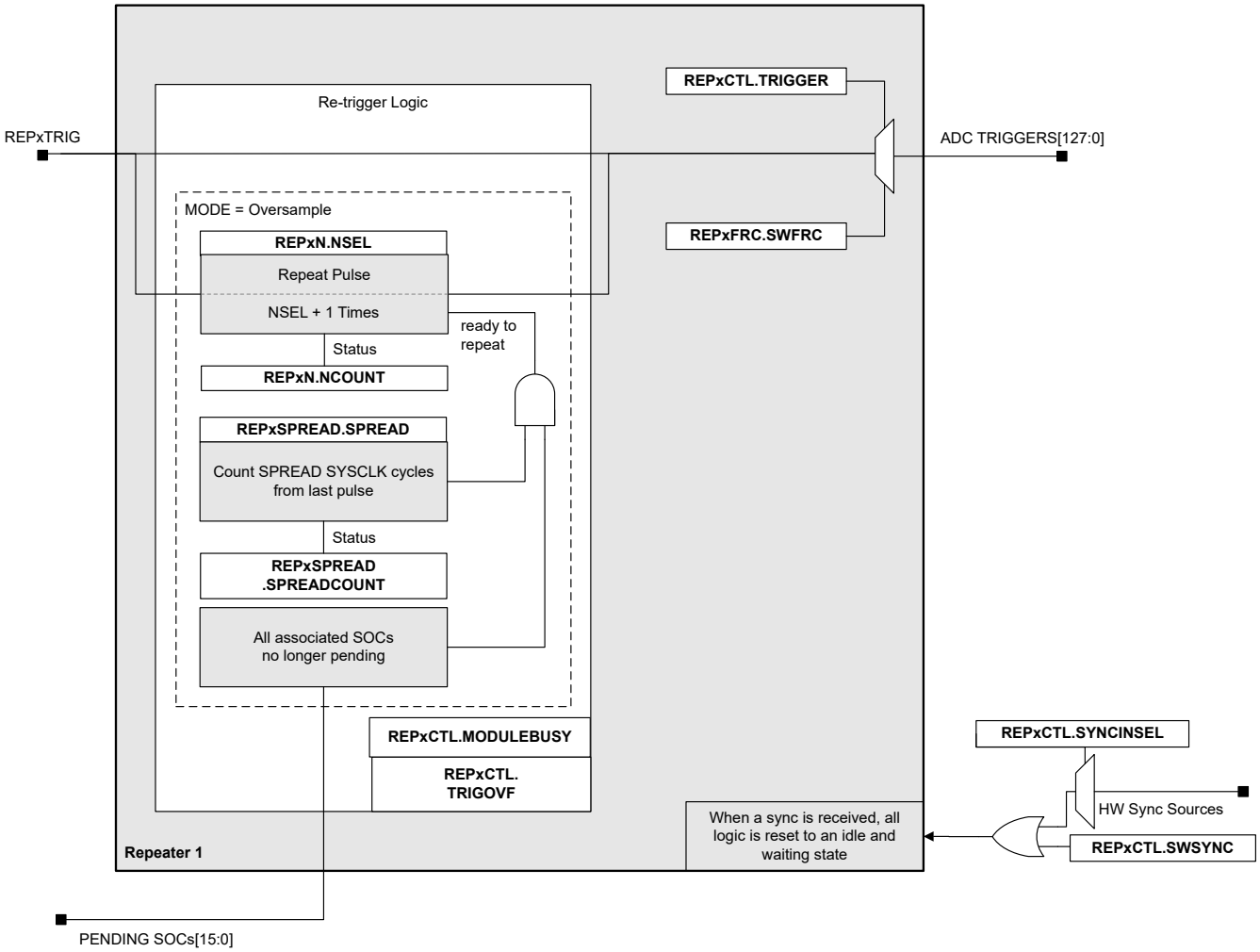
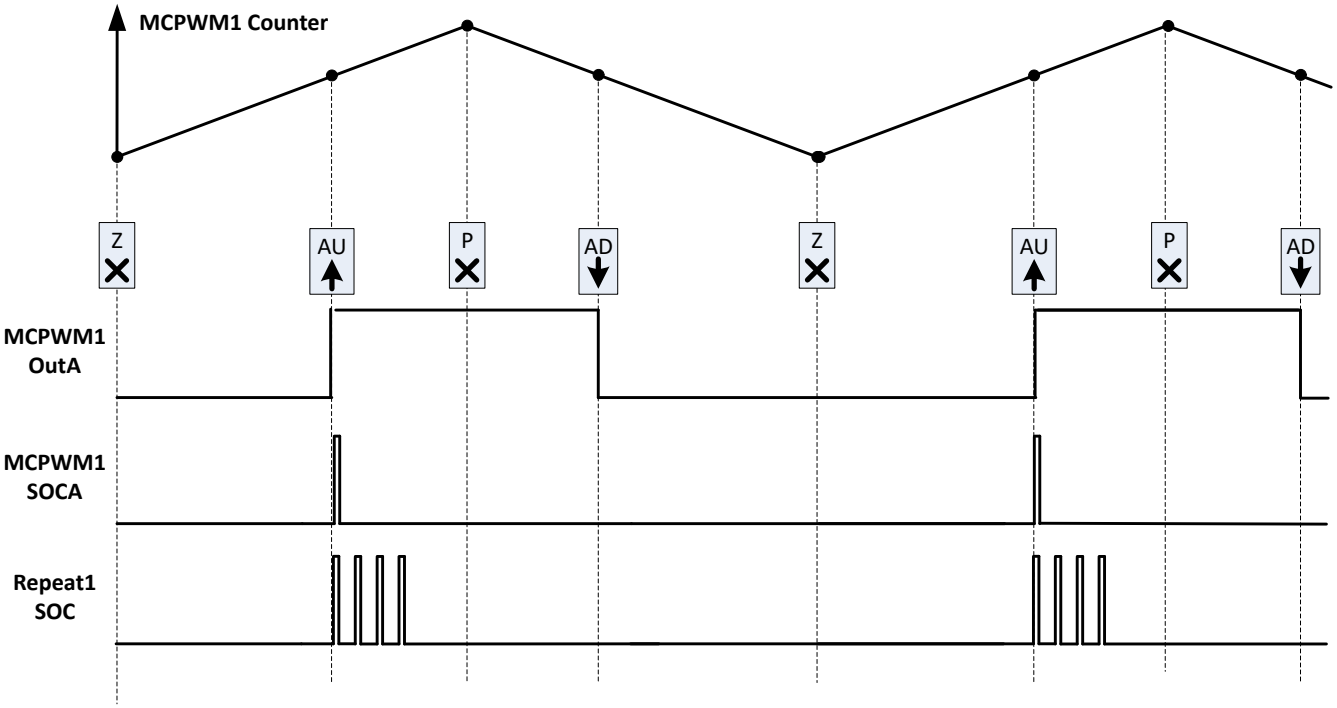


Figure 12-3. ADC Trigger Repeater Block Diagram

12.3.2.1.1 Oversampling Mode

In this mode, the repeater module passes the initial trigger through to the output. As soon as all SOCs configured to receive the trigger are in progress or completed, the repeater issues the trigger again. The process repeats until the configured number of trigger pulses (NSEL + 1) have been issued. The trigger can be repeated up to 8 times.

This mode allows the application to easily perform multiple back-to-back samples from a single trigger pulse. When used in conjunction with the aggregation options in the post-processing block, this mode enables oversampling, averaging, or peak detection. Figure 12-4 shows an example of oversampling SOCs generated from a single MCPWM trigger.



TRIGGER = MCPWM SOCA, NSEL = 3, SPREAD = 0

Figure 12-4. Oversampled ADC Trigger Example

12.3.2.1.2 Re-trigger Spread

If additional time between samples is desired, the application can configure SPREAD equal to the number of SYSCLK cycles desired between samples.

- By default, SPREAD = 0, and samples are re-triggered as soon as all associated SOCs are no longer pending.
- If SPREAD is set to a value smaller than the time needed for the associated SOCs to complete, then the ADC performs the triggered conversions back-to-back, and SPREAD is effectively 0.
- SPREAD has no effect in undersampling mode, or when NSEL = 0.

12.3.2.1.3 Trigger Repeater Configuration

To configure ADC oversampling using the trigger repeater module, follow this procedure:

1. Set up the SOC by writing to ADCSOCxCTL. Specify the repeater module (REP1TRIG) as the trigger source.
2. Configure the repeater module by writing to the REP1CTL register:
   a. Specify the desired SOC trigger source in the TRIGGER field.
   b. If desired, configure a sync source for the repeater module in the SYNCINSEL field. A sync event resets all repeater registers to a ready and waiting state, while preserving NSEL. A software-initiated sync is also possible by writing 1 to the SWSYNC bit.
   c. If desired, clear any previously set phase and trigger overflow flags by writing to the TRIGGEROVF bits.
3. Configure the trigger repeat count by writing to the REP1N.NSEL register. The repeater module supports up to 8 repeats for each trigger.
4. To configure a re-trigger spread delay in oversampling mode, write the desired delay value in SYSCLK cycles to the REP1SPREAD.SPREAD register.
5. Configure the PPB1LIMIT register. This register defines how many samples the post-processing block accumulates before loading the partial sum value in ADCPPB1PSUM into ADCPPB1SUM.
6. The post-processing block (PPB) and trigger repeater module have independent sync source configurations. To configure the PPB sync source, write to the ADCPPB1CONFIG2 register. For more information on how to configure the ADC post-processing block, see Section 12.7.

Note

When NSEL = 0, the repeater module essentially acts as a pass-through for SOC triggers, but is still useful for applying phase delay. SOC triggers are passed through , even if there are still pending SOCs. In this scenario, the ADC sets a trigger overflow flag for the individual SOC (ADCSOCOVF1.SOCx), not the repeater module. When oversampling with NSEL > 0, the ADC sets the oversampled trigger overflow flag (REP1CTL.TRIGGEROVF) if a trigger arrives while there are pending SOCs.

When NSEL = 0, the repeater module does not set the REP1CTL.MODULEBUSY indicator. In this scenario, the application must make sure that all associated SOC flags have completed before enabling oversampling mode by setting NSEL > 0.

Note

Only the PPB1 instance can be configured for accumulation.

12.3.2.1.3.1 Register Shadow Updates

To avoid latency or processing delays between triggers, the application can write updated values to the NSEL and SPREAD registers while the repeater module is still actively working. When a new SOC trigger is received, these values are loaded into the NCOUNT and SPREADCOUNT registers, which then count down to zero as each SOC is triggered by the repeater module.

12.3.2.1.4 Re-Trigger Logic

The repeater module determines when to re-trigger based on the values of ADCSOCxCTL.TRIGSEL and the SOC flags in ADCSOCFLG1. A repeat trigger is issued when all SOCs configured to be triggered by the repeater module instance are no longer pending. Figure 12-5 describes the trigger repeat logic. Because the SOC pending flag goes low at the end of the sample and hold phase, the module has plenty of time to re-trigger conversions without introducing any latency between repeat conversions.

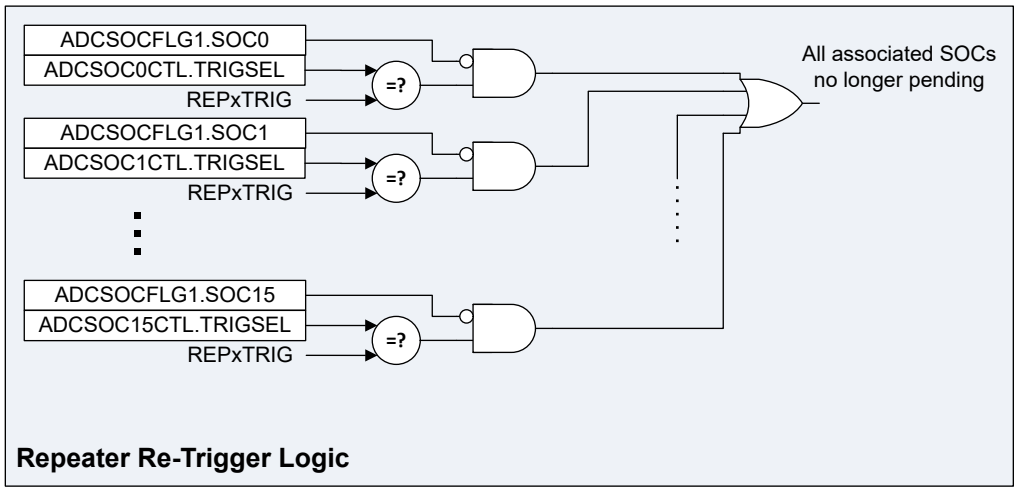


Figure 12-5. Trigger Repeater Repeat Logic

12.3.2.1.5 Multi-Path Triggering Behavior

With the trigger repeater modules, to have one trigger source take multiple paths to set SOCs in various ways is possible. For example, MCPWM1 can directly trigger SOC3 and SOC4, while one repeater block uses MCPWM1 to generate oversampling triggers on SOCs 0-2, and the second repeater block generates oversampled triggers to SOC5. Assuming all SOCs are configured for round-robin priority and the various triggers all arrive in the same cycle, the conversion order is SOC0 to SOC5 in increasing order; this is then followed by the oversampled conversions on SOC0-SOC2.

12.3.3 ADC Acquisition (Sample and Hold) Window

External signal sources vary in the ability to drive an analog signal quickly and effectively. To achieve rated resolution, the signal source needs to charge the sampling capacitor in the ADC core to within 0.5 LSBs of the signal voltage. The acquisition window is the amount of time the sampling capacitor is allowed to charge and is configurable per SOCx by the ADCSOCxCTL.ACQPS register.

ACQPS is a 8-bit register field that can be set to a value between 0 and 255. The 2 upper bits (ADCSOCxCTL.ACQPS[7:6]) configure the base duration and cycle prescalers. The base duration can be set to values of 0, 64, 192, or 448 with corresponding cycle prescalers of 1, 2, 4, and 16 respectively. The lower 6 bits (ADCSOCxCTL.ACQPS[5:0]) in the register field configure the additional cycles to be multiplied by the prescaler and added to the base duration.

Acquisition Window [SYSCLK cycles] = (Base duration) + ((Additional cycles + 1)*(Prescaler))

- The acquisition window duration is based on the System Clock (SYSCLK), not the ADC clock (ADCCLK).
- The selected acquisition window duration must be at least as long as one ADCCLK cycle.
- The data sheet specifies a minimum acquisition window duration (in nanoseconds). The user is responsible for selecting an acquisition window duration that meets this requirement.

12.3.4 Sample Capacitor Reset

In certain systems where the ADC successively samples multiple signal sources, memory crosstalk can occur. Memory crosstalk is the tendency of the ADC conversion to be pulled towards the value of the previous conversion, due to inadequate acquisition/settling time. This happens because the ADC sample capacitor voltage starts near the previously converted voltage, then settles towards the newly applied voltage on the current channel. If the acquisition window is not long enough for the sample capacitor to settle, this can result in some sample error reflected in the ADC conversion.

The 12-bit ADC modules in this device include a sample capacitor reset feature to help mitigate memory crosstalk. When sample capacitor reset is enabled, after every conversion, the sampling capacitor voltage is reset to either the VREFLO or VREFHI/2 voltage, depending on the configuration of the ADCSOCxCTL.SAMPCAPRESETSEL bit. This reset takes an extra ADCCLK cycle to complete. The sample capacitor reset function is inactive by default for each SOC. If desired, the application can activate sample capacitor reset by writing 0 to the SAMPCAPRESETDISABLE bit in the ADCSOCxCTL register. When sample capacitor reset is active, overall ADC throughput is slightly decreased due to the extra ADCCLK cycle in the conversion period.

12.3.5 ADC Input Models

For single-ended operation, the ADC input characteristics for values in the single-ended input model (see Figure 12-6) can be found in the device data sheet.

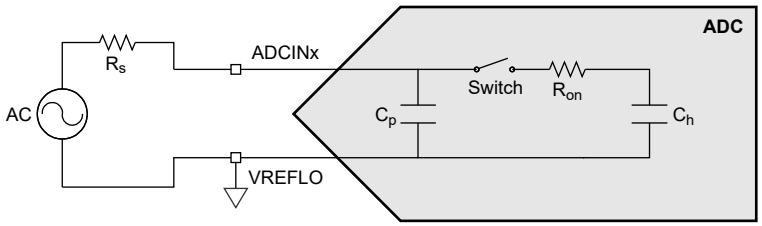


Figure 12-6. Single-Ended Input Model

These input models must be used along with actual signal source impedance to determine the acquisition window duration. See Section 12.12.1 for more information.

12.3.6 Channel Selection

Each SOC can be configured to convert any of the ADC channels. This behavior is selected for SOCx by the ADCSOCxCTL.CHSEL register. This is summarized in Table 12-5.

Table 12-5. Channel Selection of Input Pins

Input Mode	CHSEL	Input
Single-Ended	0	ADCIN0
	1	ADCIN1
	2	ADCIN2
	3	ADCIN3
	4	ADCIN4
	5	ADCIN5
	6	ADCIN6
	7	ADCIN7
	8	ADCIN8
	9	ADCIN9
	10	ADCIN10
	11	ADCIN11
	12	ADCIN12
	13	ADCIN13
	14	ADCIN14
	15	ADCIN15

12.4 SOC Configuration Examples

The following sections provide some specific examples of how to configure the SOCs to produce some conversions.

12.4.1 Single Conversion from MCPWM Trigger

To configure ADCA to perform a single conversion on channel ADCIN1 when the MCPWM timer reaches the period match, a few things are necessary. First, MCPWM3 must be configured to generate an SOCA, SOCB, SOCC, or SOCD signal (in this statement, SOC refers to a signal in the MCPWM module). See the *Multi-Channel Pulse Width Modulator Module (MCPWM)* chapter on how to do this. Assume that SOCB was chosen.

SOC5 is chosen arbitrarily. Any of the 16 SOCs can be used.

Assuming a 100ns sample window is desired with a SYSCLK frequency of 160MHz, then the acquisition window duration must be 100ns/6.25ns = 16 cycles. The ACQPS field must be set to 0 in the upper two bits and 16 - 1 in the lower six bits (ACQPS[7:6] = 0, ACQPS[5:0]=15)..

```
AdcaRegs.ADCSOC5CTL.bit.CHSEL = 1;      //SOC5 converts ADCINA1
               AdcaRegs.ADCSOC5CTL.bit.ACQPS = 15;     //SOC5 uses a sample duration of 16 SYSCLK
cycles
AdcaRegs.ADCSOC5CTL.bit.TRIGSEL = 14;   //SOC5 begins conversion on PWM3 SOCB
```

As configured, when MCPWM3 matches the period and generates the SOCB signal, the ADC begins sampling channel ADCINA1 (SOC5) immediately if the ADC is idle. If the ADC is busy, ADCINA1 begins sampling when SOC5 gains priority (see Section 12.5). The ADC control logic samples ADCINA1 with the specified acquisition window width of 100ns. Immediately after the acquisition is complete, the ADC begins converting the sampled voltage to a digital value. When the ADC conversion is complete, the results are available in the ADCRESULT5 register (see Section 12.11 for exact sample, conversion, and result latch timings).

12.4.2 Multiple Conversions from CPU Timer Trigger

This example shows how to sample multiple signals with different acquisition window requirements. CPU1 Timer 2 is used to generate the trigger. To see how to configure the CPU timer, see the *System Control and Interrupts* chapter.

A good first step when designing a sampling scheme with many signals is to list out the signals and the required acquisition window. From this, calculate the necessary number of SYSCLK cycles for each signal, then the ACQPS register setting. This is shown in Table 12-6, where a SYSCLK of 160MHz is assumed (6.25ns cycle time).

Table 12-6. Example Requirements for Multiple Signal Sampling

Signal Name	Acquisition Window Requirement	Acquisition Window (SYSCLK Cycles)	ACQPS Register Value
Signal 1	>160ns	160ns/6.25ns = 26 (round up)	26 – 1 = 25
Signal 2	>592ns	592ns/6.25ns = 94 (round down)	0x01000000 | ((94 - 64) / 2) - 1 = 14
Signal 3	>147ns	147ns/6.25ns = 24 (round up)	24 – 1 = 23
Signal 4	>392ns	392ns/6.25ns = 63 (round up)	63 – 1 = 62

Next decide which ADC pins to connect to each signal. This is highly dependent on the application board layout. Once the pins are selected, determining the value of CHSEL is straightforward (see Table 12-7).

Table 12-7. Example Connections for Multiple Signal Sampling

Signal Name	ADC Pin	CHSEL Register Value
Signal 1	ADCINA5	5
Signal 2	ADCINA0	0
Signal 3	ADCINA3	3
Signal 4	ADCINA2	2

With the information tabulated, generate the SOC configurations:

```
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 5;        //SOC0 converts ADCINA5
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 23;       //SOC0 uses a sample duration of 24 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 3;      //SOC0 begins conversion on CPU1 Timer 2
AdcaRegs.ADCSOC1CTL.bit.CHSEL = 0;        //SOC1 converts ADCINA0
AdcaRegs.ADCSOC1CTL.bit.ACQPS = 88;       //SOC1 uses a sample duration of 89 SYSCLK cycles
AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 3;      //SOC1 begins conversion on CPU1 Timer 2
AdcaRegs.ADCSOC2CTL.bit.CHSEL = 3;        //SOC2 converts ADCINA3
AdcaRegs.ADCSOC2CTL.bit.ACQPS = 21;       //SOC2 uses a sample duration of 22 SYSCLK cycles
AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 3;      //SOC2 begins conversion on CPU1 Timer 2
AdcaRegs.ADCSOC3CTL.bit.CHSEL = 2;        //SOC3 converts ADCINA2
AdcaRegs.ADCSOC3CTL.bit.ACQPS = 58;       //SOC3 uses a sample duration of 59 SYSCLK cycles
AdcaRegs.ADCSOC3CTL.bit.TRIGSEL = 3;      //SOC3 begins conversion on CPU1 Timer 2
```

As configured, when CPU1 Timer 2 generates an event, SOC0, SOC1, SOC2, and SOC3 eventually is sampled and converted, in that order. The conversion results for ACINA5 (Signal 1) are in ADCRESULT0. Similarly, The results for ADCINA0 (Signal 2), ADCINA3 (Signal 3), and ADCINA2 (Signal 4) are in ADCRESULT1, ADCRESULT2, and ADCRESULT3, respectively.

Note

There is a possibility, but unlikely, that the ADC can begin converting SOC1, SOC2, or SOC3 before SOC0 depending on the position of the round-robin pointer when the CPU Timer trigger is received. See Section 12.5 to understand how the next SOC to be converted is chosen.

12.4.3 Software Triggering of SOCs

At any point, whether or not the SOCs have been configured to accept a specific trigger, a software trigger can set the SOCs to be converted. This is accomplished by writing bits in the ADCSOCFRC1 register.

Software triggering of the previous example without waiting for the CPU1 Timer 2 to generate the trigger can be accomplished by the statement:

```
AdcaRegs.ADCSOCFRC1.all = 0x000F;        //set SOC flags for SOC0 to SOC3
```

12.5 ADC Conversion Priority

When multiple SOC flags are set at the same time, one of two forms of priority determines the converted order. The default priority method is round-robin. In this scheme, no SOC has an inherent higher priority than another. Priority depends on the round-robin pointer (RRPOINTER). The RRPOINTER reflected in the ADCSOCPRIORITYCTL register points to the last SOC converted. The highest priority SOC is given to the next value greater than the RRPOINTER value, wrapping around back to SOC0 after SOC15. At reset the value is 16 since 0 indicates a conversion has already occurred. When RRPOINTER equals 16 the highest priority is given to SOC0. The RRPOINTER is reset when the ADC module is reset or when the reset value is written to the SOCPRICTL register. The ADC module is reset by writing and clearing the SOFTPRES bit corresponding to the ADC instance.

An example of the round-robin priority method is given in Figure 12-7.

The SOCPRIORITY field in the ADCSOCPRIORITYCTL register can be used to assign high priority from a single to all of the SOCs. When configured as high priority, an SOC interrupts the round-robin wheel after any current conversion completes and inserts in as the next conversion. After the conversion completes, the round-robin wheel continues where the conversion was interrupted. If two high priority SOCs are triggered at the same time, the SOC with the lower number takes precedence.

High priority mode is assigned first to SOC0, then in increasing numerical order. The value written in the SOCPRIORITY field defines the first SOC that is not high priority. In other words, if a value of 4 is written into SOCPRIORITY, then SOC0, SOC1, SOC2, and SOC3 are defined as high priority, with SOC0 the highest.

An example using high priority SOC's is given in Figure 12-8.

A After reset, SOC0 is highest priority SOC; SOC7 receives trigger; SOC7 configured channel is converted immediately.

B RRPOINTER changes to point to SOC 7; SOC8 is now highest priority SOC.

C SOC2 & SOC12 triggers rcvd. simultaneously; SOC12 is first on round robin wheel; SOC12 configured channel is converted while SOC2 stays pending.

D RRPOINTER changes to point to SOC 12; SOC2 configured channel is now converted.

E RRPOINTER changes to point to SOC 2; SOC3 is now highest priority SOC.

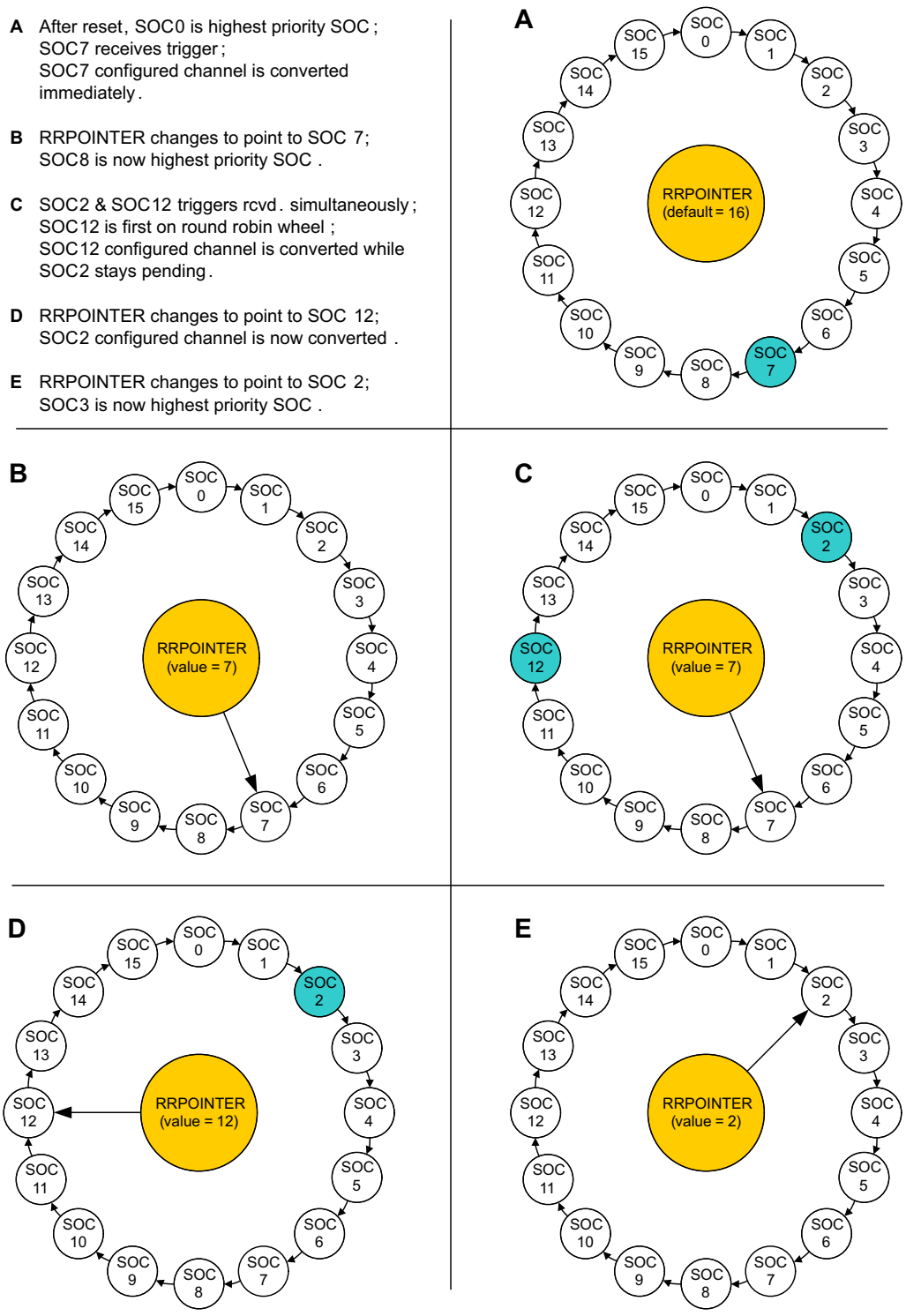


Figure 12-7. Round Robin Priority Example

Example when SOCPRIORITY = 4

A After reset, SOC4 is 1st on round robin wheel ;
SOC7 receives trigger ;
SOC7 configured channel is converted immediately .

B RRPOINTER changes to point to SOC 7;
SOC8 is now 1st on round robin wheel .

C SOC2 & SOC12 triggers rcvd . simultaneously ;
SOC2 interrupts round robin wheel and SOC 2 configured channel is converted while SOC 12 stays pending .

D RRPOINTER stays pointing to 7;
SOC12 configured channel is now converted .

E RRPOINTER changes to point to SOC 12;
SOC13 is now 1st on round robin wheel .

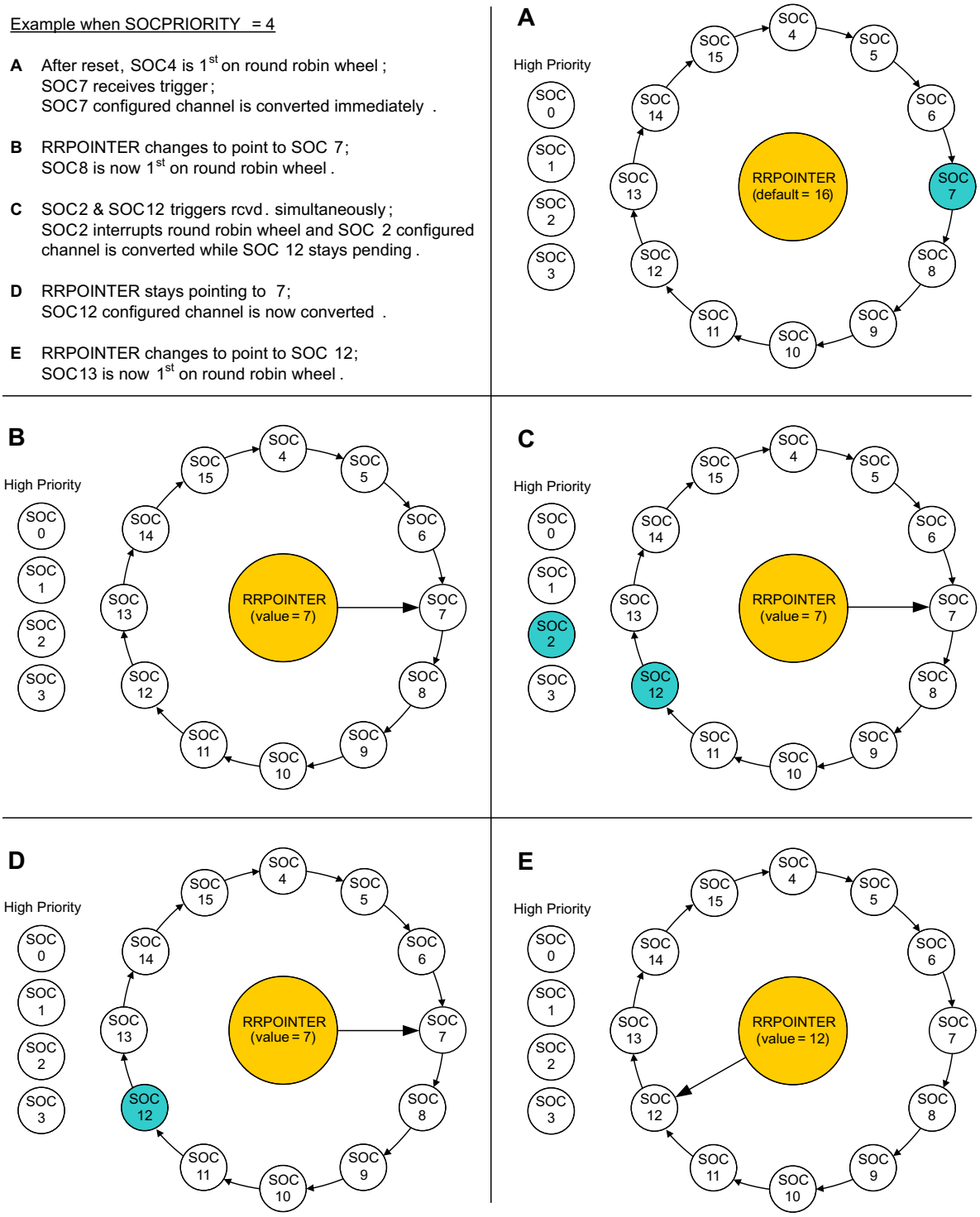


Figure 12-8. High Priority Example

12.6 EOC and Interrupt Operation

Each SOC has a corresponding end-of-conversion (EOC) signal. This EOC signal can be used to trigger an ADC interrupt. The ADC can be configured to generate the EOC pulse at either the end of the acquisition window or at the end of the voltage conversion. This is configured using the bit INTPULSEPOS in the ADCCTL1 register. See Section 12.11 for exact EOC pulse location.

The ADC module has 2 configurable ADC interrupts. These interrupts can be triggered by any of the 16 EOC signals. The flag bit for each ADCINT can be read directly to determine if the associated SOC is complete or the interrupt can be passed on to the PIE.

It is also possible to generate an ADC interrupt based on a PPB oversampling logic event on PPB1, such as when the sample count matches the configured limit. There is one oversampling interrupt (OSINT) flag available in the module for this purpose. Any of the ADCINT flags can be configured for an OSINT by configuring the INTxSEL field the corresponding ADCINTSELxNy register.

Note

The ADCCTL1.ADCBSY bit being clear does not indicate that all conversions in a set of SOCs have completed, only that the ADC is ready to process the next conversion. To determine if a sequence of SOCs is complete, link an ADCINT flag to the last SOC in the sequence and monitor that ADCINT flag.

Figure 12-9 shows a block diagram of the ADC interrupt structure.

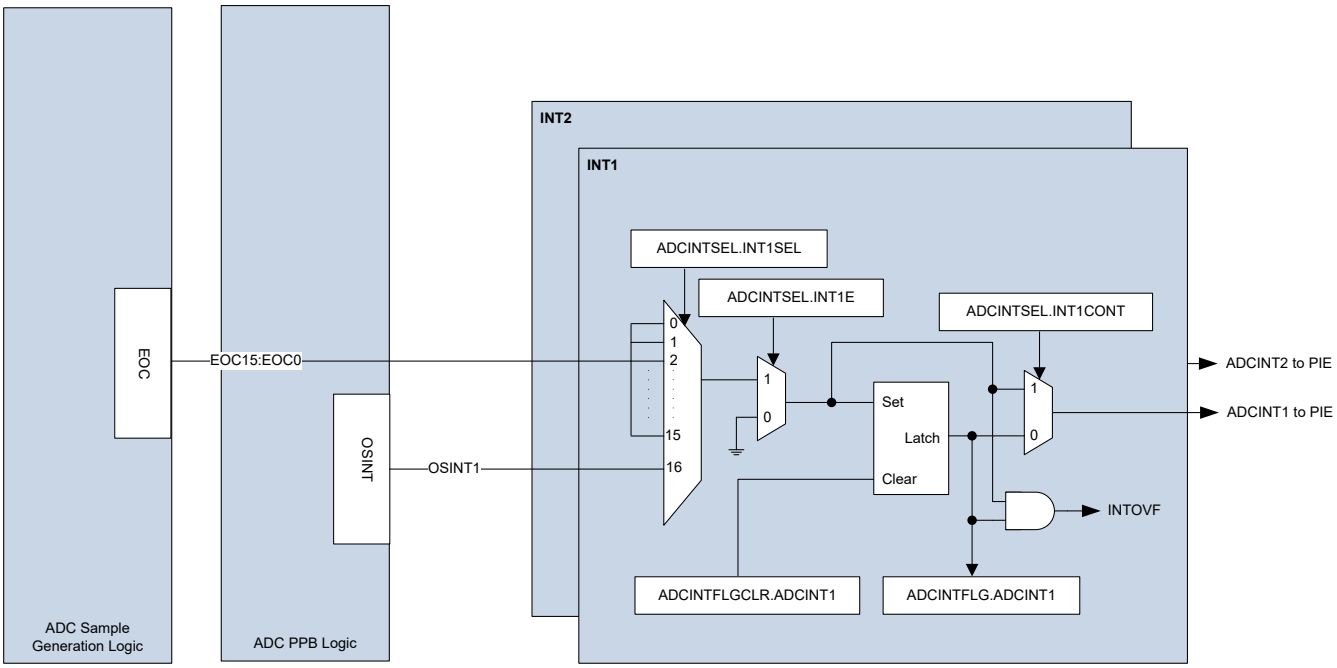


Figure 12-9. ADC End-of-Conversion (EOC) Signal Interrupts

12.6.1 Interrupt Overflow

If the EOC signal sets a flag in the ADCINTFLG register, but that flag is already set, an interrupt overflow occurs. By default, overflow interrupts are not passed on to the module. When an overflow occurs on a given flag in the ADCINTFLG register, the corresponding flag in the ADCINOVF register is set. This overflow flag is only used to detect that an overflow has occurred; the flag does not block further interrupts from propagating to the module.

When an ADC interrupt overflow occurs, the application must check the appropriate ADCINTOVF flag inside the ISR or in the background loop and take appropriate action when an overflow is detected. The following code snippets demonstrate how to check the ADCINTOVF flag inside the ISR after attempting to clear the ADCINT flag.

```
// Clear the interrupt flag
AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;      //clear INT1 flag for ADC-A

// Check if an overflow has occurred
if(1 == AdcaRegs.ADCINTOVF.bit.ADCINT1)     //ADCINT overflow occurred
{
    AdcaRegs.ADCINTOVFCLR.bit.ADCINT1 = 1   //Clear overflow flag
    AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1   //Re-clear ADCINT flag
}
```

```
//
// Clear the interrupt flag
//
ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);

//
// Check if an overflow has occurred
//
if(true == ADC_getInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1))
{
    ADC_clearInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1);
    ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
}
```

12.6.2 Continue to Interrupt Mode

The INTxCONT bits in the ADCINTSEL1N2 register configures how interrupts are handled when an ADCINTFLG has not yet been cleared from a prior interrupt. This mode is disabled by default and additional overlapping interrupts are not issued to the PIE. By activating this mode, ADC interrupts always reach the PIE. If interrupts occur while ADCINTFLG is set, the ADCINTOVF register remains set regardless of the configuration of the INTxCONT bits.

12.6.3 Early Interrupt Configuration Mode

Enabling early interrupt mode can allow the application to enter the ADC interrupt service routine before the ADC results are ready. This allows the application to do any necessary pre-work so that the application can act on the ADC results immediately when the ADC results become available. If the timing of the early interrupt is too early, then the application needs to waste time until the updated ADC results become available. To prevent this situation, the time the ADC interrupt is entered in early interrupt mode is configurable by way of the DELAY field in the ADCINTCYCLE register.

- To use the configurable interrupt time, the ADC must be in early interrupt mode. To achieve this, clear the bit INTPULSEPOS to 0 in ADCCTL1.
- The DELAY value in the ADCINTCYCLE register sets the number of additional SYSCLK cycles after the falling edge of the SOC pulse before the ADCINT flag is set.
- If the value of DELAY goes beyond EOC, the ADC interrupt is generated along with EOC.
- Writing values to DELAY when INTPULSEPOS is set to 1 does not have any effect on the interrupt generation.

12.7 Post-Processing Blocks

Each ADC module contains three post-processing blocks (PPB). These blocks can be associated with any of the 16 RESULT registers using the ADCPPBxCONFIG.CONFIG bit field. The post-processing blocks have the ability to:

- Remove an offset associated with the ADCIN channel
- Subtract out a reference value
- Flag a zero-crossing point, with the option to trip a PWM and generate an interrupt
- Flag a high or low compare limit, with the option to trip a PWM and generate an interrupt
- Flag an in-limit compare limit, with the option to trip a PWM and generate an interrupt

Figure 12-10 presents the structure of each PPB. Subsequent sections explain the use of each submodule.

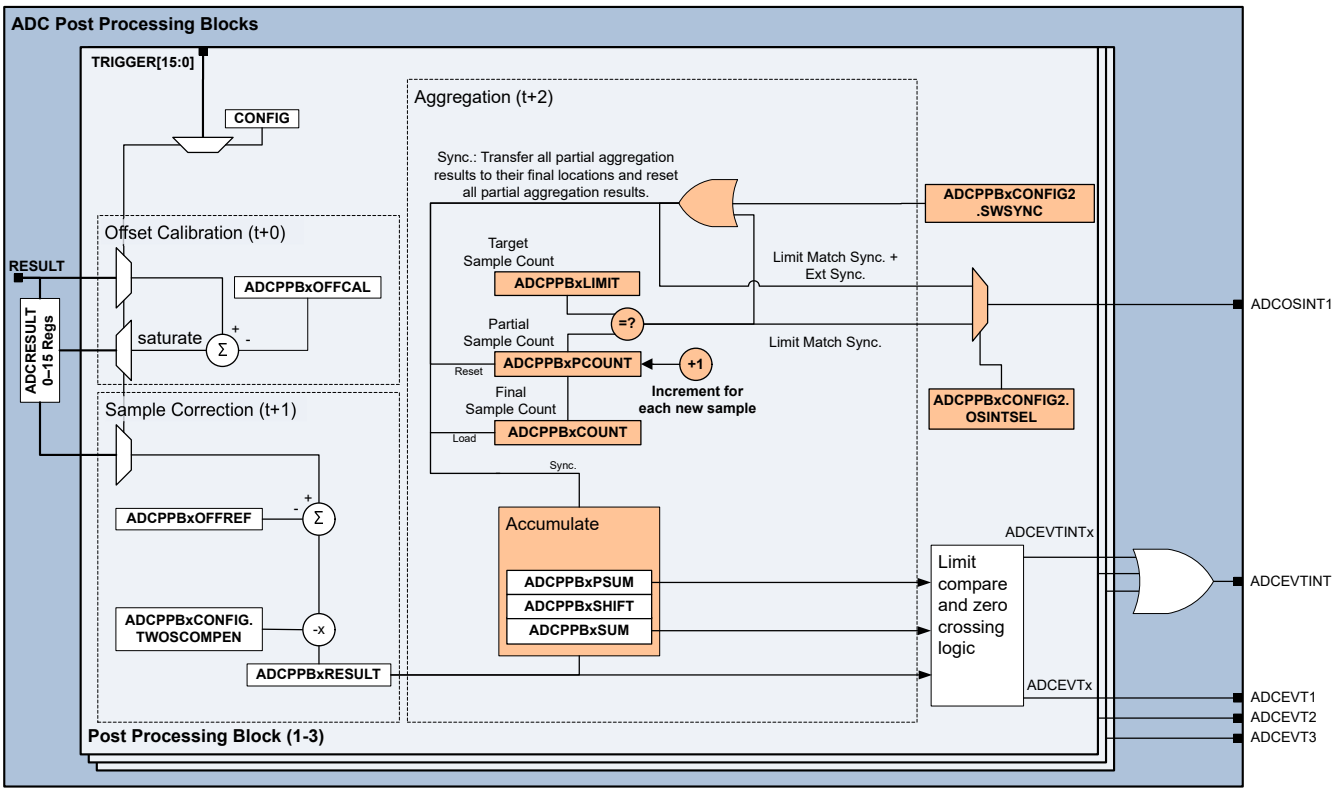


Figure 12-10. ADC Post-Processing Blocks (PPB) Block Diagram

Note

There are three PPB instances available in this device. The highlighted blocks in Figure 12-10 are supported only in one of the PPB instances.

12.7.1 PPB Offset Correction

In many applications, external sensors and signal sources produce an offset. A global trimming of the ADC offset is not enough to compensate for these offsets, which vary from channel to channel. The post-processing block can remove these offsets with zero overhead, saving numerous cycles in tight control loops.

Offset correction is accomplished by first pointing the ADCPPBxCONFIG.CONFIG to the desired SOC, then writing an offset correction value to the ADCPPBxOFFCAL.OFFCAL register. The post-processing block automatically adds or subtracts the value in the OFFCAL register from the raw conversion result and stores the value in the ADCRESULT register. This addition/subtraction saturates at 0 on the low end and 4095 on the high end.

Note

- Writing a 0 to the OFFCAL register effectively disables the offset correction feature, passing the raw result unchanged to the ADCRESULT register.
- To point multiple PPBs to the same SOC is possible. In this case, the OFFCAL value that is actually applied comes from the PPB with the number.

12.7.2 PPB Error Calculation

In many applications, an error from a set point or expected value must be computed from the digital output of an ADC conversion. In other cases, a bipolar signal is necessary or convenient for control calculations. The PPB can perform these functions automatically, reducing the sample to output latency and reducing software overhead.

Error calculation is accomplished by first pointing the ADCPPBxCONFIG.CONFIG to the desired SOC, then writing a value to the ADCPPBxOFFCAL.OFFREF register. The post-processing block automatically subtracts the value in the OFFREF register from the ADCRESULT value and stores the value in the ADCPPBxRESULT register. This subtraction produces a sign-extended 32-bit result. It is also possible to selectively invert the calculated value before storing in the ADCPPBxRESULT register by setting the TWOSCOMPEN bit in the ADCPPBxCONFIG register.

Note

- Do not write a value larger than 12 bits to the ADCPPBxOFFREF register.
- Since the ADCPPBxRESULT register is unique for each PPB, to point multiple PPBs to the same SOC and get different results for each PPB is possible.
- Writing a 0 to the ADCPPBxOFFREF register effectively disables the error calculation feature, passing the ADCRESULT value unchanged to the ADCPPBxRESULT register.

12.7.3 PPB Limit Detection and Zero-Crossing Detection

Many applications perform a limit check against the ADC conversion results. The PPB can automatically perform a check against a high limit, a low limit, high and low limits simultaneously, or whenever ADCPPBxRESULT changes sign. Based on these comparisons, the PPB can generate a trip to the PWM and an interrupt automatically, lowering the sample to MCPWM latency and reducing software overhead. This functionality also enables safety-conscious applications to trip the MCPWM based on an out-of-range ADC conversion without any CPU intervention.

To enable this functionality, first point the ADCPPBxCONFIG.CONFIG to the desired SOC, then write a value to one or both of the registers ADCPPBxTRIPHI.LIMITHI and ADCPPBxTRIPLO.LIMITLO (zero-crossing detection does not require further configuration). Whenever these limits are exceeded, the PPBxTRIPHI bit or PPBxTRIPLO bit is set in the ADCEVTSTAT register. When the PPB result is either in between or equal to the PPBxTRIPHI and PPBxTRIPLO thresholds, the PPBxINLIMIT bit is set. Note that the PPBxZERO bit in the ADCEVTSTAT register is gated by end-of-conversion (EOC), not by the sign change in the ADCPPBxRESULT register. The ADCEVTCLR register has corresponding bits to clear these event flags. The ADCEVTSEL register has corresponding bits that allow the events to propagate through to the PWM. The ADCEVTINTSEL register has corresponding bits that allow the events to propagate through to the .

One interrupt is shared between all the PPBs for a given ADC module as shown in Figure 12-11.

Note

- If different actions need to be taken for different PPB events from the same ADC module, then the ADCEVTINT ISR has to read the PPB event flags in the ADCEVTSTAT register to determine which event caused the interrupt.
- If different MCPWM trips need to be generated separately for high compare, low compare, in-limit compare, or zero-crossing, this can be achieved by pointing multiple PPBs to the same SOC.
- The zero-crossing detect circuit considers a result of zero to be positive.

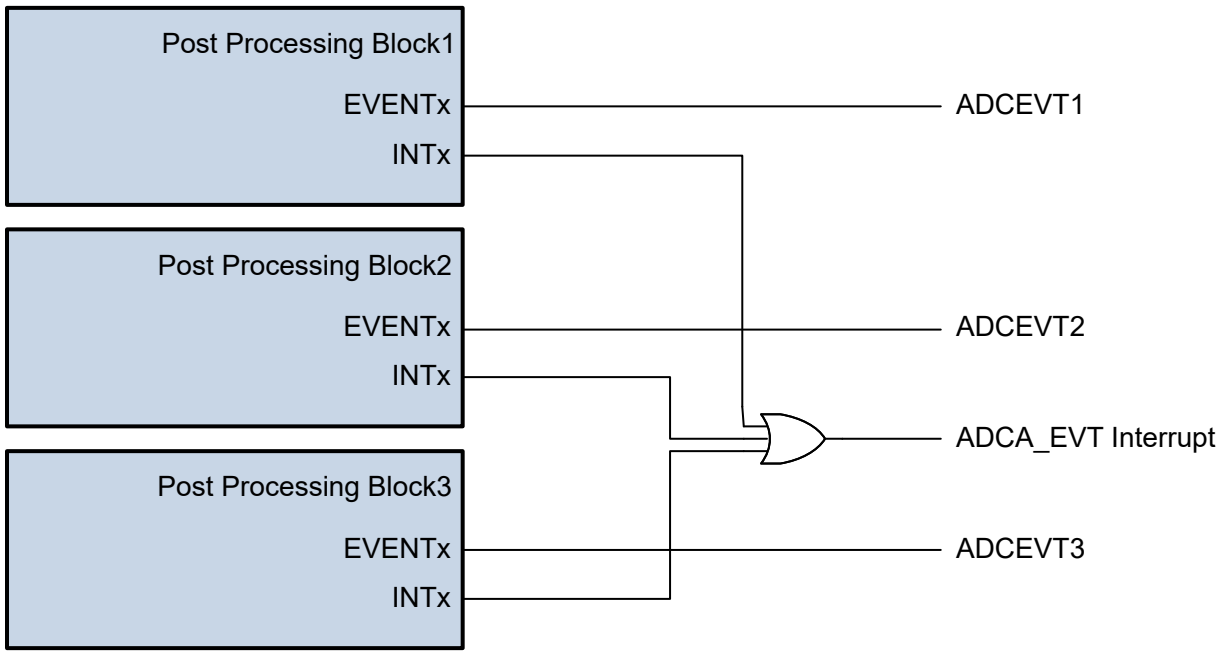


Figure 12-11. ADC Post-Processing Blocks (PPB) Interrupt Event

12.8 Opens/Shorts Detection Circuit (OSDETECT)

The opens/shorts detection circuit (OSDETECT) can be used to detect pin faults in the system. The circuit connects to the ADC input after the channel select multiplexer but before the S+H circuit as shown in Figure 12-12.

Note

- The divider resistance tolerances can vary widely; hence, this feature must not be used to check for conversion accuracy.
- See the data sheet for implementation and availability of analog input channels.
- Due to high drive impedance, a S+H duration much longer than the ADC minimum is needed.

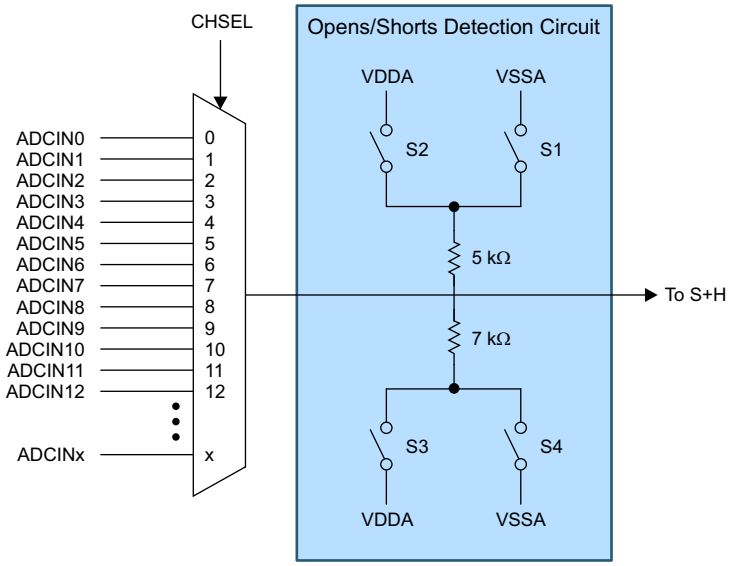


Figure 12-12. Opens/Shorts Detection Circuit

The circuit can be operated by writing a value to the DETECTCFG field in the ADCOSDETECT register from the Analog Subsystem. This causes the circuit to source a voltage onto the input during the S+H phase of any conversion. The voltage and drive strength of the OSDETECT circuit for different DETECTCFG settings is given in Table 12-8.

Table 12-8. DETECTCFG Settings

ADCOSDETECT. DETECTCFG	Source Voltage	S4	S3	S2	S1	Drive Impedance
0	Off	Open	Open	Open	Open	Open
1	Zero Scale	Closed	Open	Open	Closed	5K || 7K
2	Full Scale	Open	Closed	Closed	Open	5K || 7K
3	5/12 VDDA	Open	Closed	Open	Closed	5K || 7K
4	7/12 VDDA	Closed	Open	Closed	Open	5K || 7K
5	Zero Scale	Open	Open	Open	Closed	5K
6	Full Scale	Open	Open	Closed	Open	5K
7	Zero Scale	Closed	Open	Open	Open	7K

Additionally, configure the TESTANA1_CONFIG field in the register to enable TESTANA1.

12.8.1 Open Short Detection Implementation

A representative circuit with the Open-short Detection (OSDETECT) implementation consists of the signal source with series resistance R_S, shunt capacitor C_P, the equivalent OSDETECT resistance $R_{OSDETECT}$ and voltage $V_{OSDETECT}$ is shown in Figure 12-13 and can be used as a basis to calculate the signal level going in to the sampling capacitor. $R_{OSDETECT}$ and $V_{OSDETECT}$ are the equivalent input resistance and voltage source contributed by the OSDETECT circuit with values shown in Table 12-8 for the different configuration settings. Refer to Figure 12-13 when deriving the input signal to S/H if signal source V_S is driving while the OSDETECT feature is enabled.

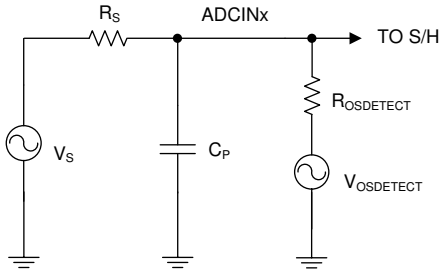


Figure 12-13. Input Circuit Equivalent with OSDETECT Enabled

The input impedance R_S and C_P are integral parts of the signal source or can have been implemented in the design to precondition the signal or to control signal settling time to meet S/H requirements. The input path has to be considered when using the OSDETECT feature, as this affects the conversion results. For instance, driving an input signal when this feature is enabled connects signal V_S to the OSDETECT circuit through R_S and affects the ADC results. Larger C_P values (in the order greater than hundreds of pF) require using higher ACQPS to make sure the signal at the input has settled prior to conversion.

To enable the circuit:

1. Configure the ADC for conversion (for example, channel, SOC, ACQPS, prescaler, trigger, and so on).
2. Set up the ADCOSDETECT register for the desired voltage divider connection as shown in Table 12-8.
3. Initiate a conversion and inspect the conversion result.

Interpret the results based on what is driving on the input side and what are the values of R_S and C_P. If the V_S signal can be disconnected from the input pin, the circuit can be used to detect open and shorted input pins as described in the following sections.

12.8.2 Detecting an Open Input Pin

By cycling through the various OSDETECT settings, the input signal is pulled towards the sourced voltages. An input with good drive strength (pin not open) is minimally affected. However, if the pin is open, the sampled voltages is close to the source voltages specified in Table 12-8.

12.8.3 Detecting a Shorted Input Pin

By cycling through the various OSDETECT settings, the input signal is pulled towards the sourced voltages. An input with finite drive strength (pin not shorted) is pulled toward each sourced voltage. However, if the pin is shorted, the signal remains at the same voltage.

12.9 Power-Up Sequence

Upon device power-up or system level reset, the ADC is powered down and disabled. When powering up the ADC, the following sequence must be used:

1. Set the desired ADC clock divider in the PRESCALE field of ADCCTL2.
2. Power up the ADC by setting the ADCPWDNZ bit in ADCCTL1.
3. Allow a delay before sampling. See the data sheet for the necessary time.

Only one delay is necessary as long as the delay occurs after the ADC has begun powering up.

12.10 ADC Calibration

During the fabrication and test process, Texas Instruments calibrates the gain, offset, and linearity of the ADCs. These trim settings are stored in TI reserved OTP memory, and can be loaded using C-callable functions.

- The Device_cal() function copies the trim values for ADC from OTP memory to the respective trim registers.

Until the appropriate factory trim is loaded, the ADC and other analog modules are not specified to operate within the data sheet specifications. Similarly, if trim values other than the factory settings are placed into the trim registers, the ADC (and other modules) is not specified to operate within the data sheet specifications.

The boot ROM calls the calibration functions, so trim values are initially populated without user intervention. However, if the trims are cleared due to a module reset or modified for some other reason, then the user must call the calibration functions (defined in the software development kit header files).

12.10.1 ADC Zero Offset Calibration

ADC offset error is determined and calibrated during factory testing. However, the user still has the option to perform offset calibration if the end application specifically requires this. This section describes how to perform offset calibration using internal VREFLO connection for single-ended operation.

Zero offset error is defined as the difference from 0 that occurs when converting a voltage at VREFLO. The zero offset error can be positive or negative. To correct this error, an adjustment of equal magnitude and opposite polarity is written into the ADCOFFTRIM register. The value contained in this register is applied before the results are available in the ADC result registers. This operation is fully contained within the ADC core, so the timing of the results is not affected, and the full dynamic range of the ADC is maintained for any trim value.

Note

Regardless of the converter resolution, the size of each ADCOFFTRIM step is (VREFHI-VREFLO)/ 65536.

Use the following procedure to re-calibrate the ADC offset in 12-bit single-ended mode:

1. Set ADCOFFTRIM to +112 steps (0x70). This adds an artificial offset to account for negative offset that can reside in the ADC core.
2. Perform some multiple of 16 conversions on VREFLO (internal connection), accumulating the results (for example, 32 × 16 conversions = 512 conversions). Use the maximum value of ACQPS to make sure longer settling time to account for parasitic impedance of internal VREFLO connections.
3. Divide the accumulated result by the multiple of 16 (for example, for 512 conversions, divide by 32).
4. Set ADCOFFTRIM to 112 – result from step 3.

12.11 ADC Timings

The process of converting an analog voltage to a digital value is broken down into an S+H phase and a conversion phase. The ADC sample and hold circuits (S+H) are clocked by SYSCLK while the ADC conversion process is clocked by ADCCLK. ADCCLK is generated by dividing down SYSCLK based on the PRESCALE field in the ADCCTL2 register.

The S+H duration is the value of the ACQPS field of the SOC being converted, plus one, times the SYSCLK period. The user must make sure that this duration exceeds both 1 ADCCLK period and the minimum S+H duration specified in the data sheet. The conversion time is approximately 10.5 ADCCLK cycles. The exact conversion time is always a whole number of SYSCLK cycles. See the timing diagrams and tables in Section 12.11.1 for exact timings.

12.11.1 ADC Timing Diagrams

The following diagrams show the ADC conversion timings for two SOCs given the following assumptions:

- SOC0 and SOC1 are configured to use the same trigger.
- No other SOCs are converting or pending when the trigger occurs.
- The round robin pointer is in a state that causes SOC0 to convert first.
- ADCINTSEL is configured to set an ADCINT flag upon end of conversion for SOC0 (whether this flag propagates through to the CPU to cause an interrupt is determined by the configurations in the PIE module).

Table 12-9 describes the parameters in the following timing diagrams.

Table 12-10 and Table 12-11 lists the ADC timings.

Table 12-9. ADC Timing Parameter Descriptions

Parameter	Description
t_{SH}	The duration of the S+H window. At the end of this window, the value on the S+H capacitor becomes the voltage to be converted into a digital value. The duration is given by (ACQPS + 1) SYSCLK cycles. ACQPS can be configured individually for each SOC, so t_{SH} is not necessarily the same for different SOCs. **Note:** The value on the S+H capacitor is captured approximately 5ns before the end of the S+H window regardless of device clock settings.
t_{LAT}	The time from the end of the S+H window until the ADC results latch in the ADCRESULTx register. If the ADCRESULTx register is read before this time, the previous conversion results are returned.
t_{EOC}	The time from the end of the S+H window until the S+H window for the next ADC conversion can begin. The subsequent sample can start before the conversion results are latched.
t_{INT}	The time from the end of the S+H window until an ADCINT flag is set (if configured). If the INTPULSEPOS bit in the ADCCTL1 register is set, t_{INT} coincides with the end of conversion (EOC) signal. If the INTPULSEPOS bit is 0, and the OFFSET field in the ADCINTCYCLE register is not 0, then there is a delay of OFFSET SYSCLK cycles before the ADCINT flag is set. This delay can be used to enter the ISR or trigger the DMA exactly when the sample is ready. If the INTPULSEPOS bit is 0, t_{INT} coincides with the end of the S+H window. If t_{INT} triggers a read of the ADC result register (directly through DMA or indirectly by triggering an ISR that reads the result), care must be taken to make sure the read occurs after the results latch (otherwise, the previous results are read).
t_{DMA}	The time from the end of the S+H window until a DMA read of the ADC conversion result is triggered.

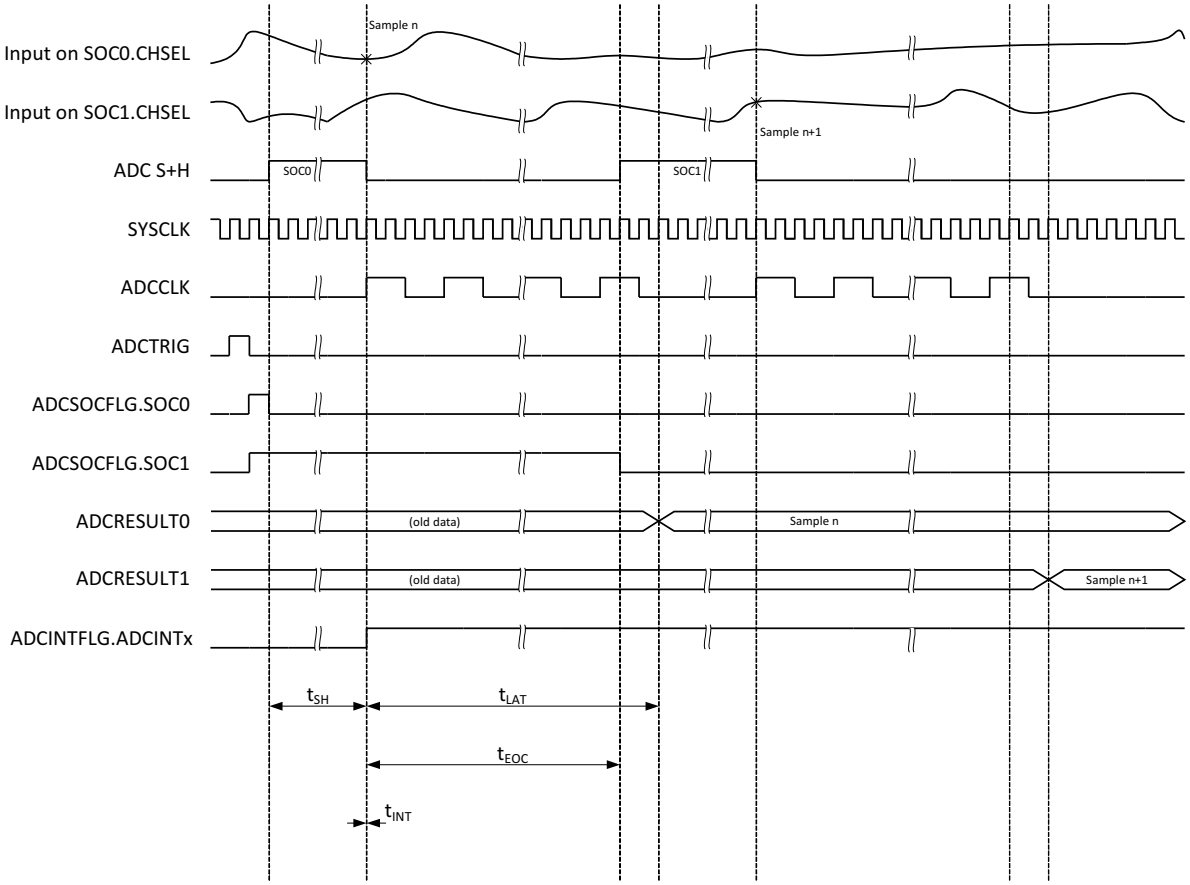


Figure 12-14. ADC Timings in Early Interrupt Mode

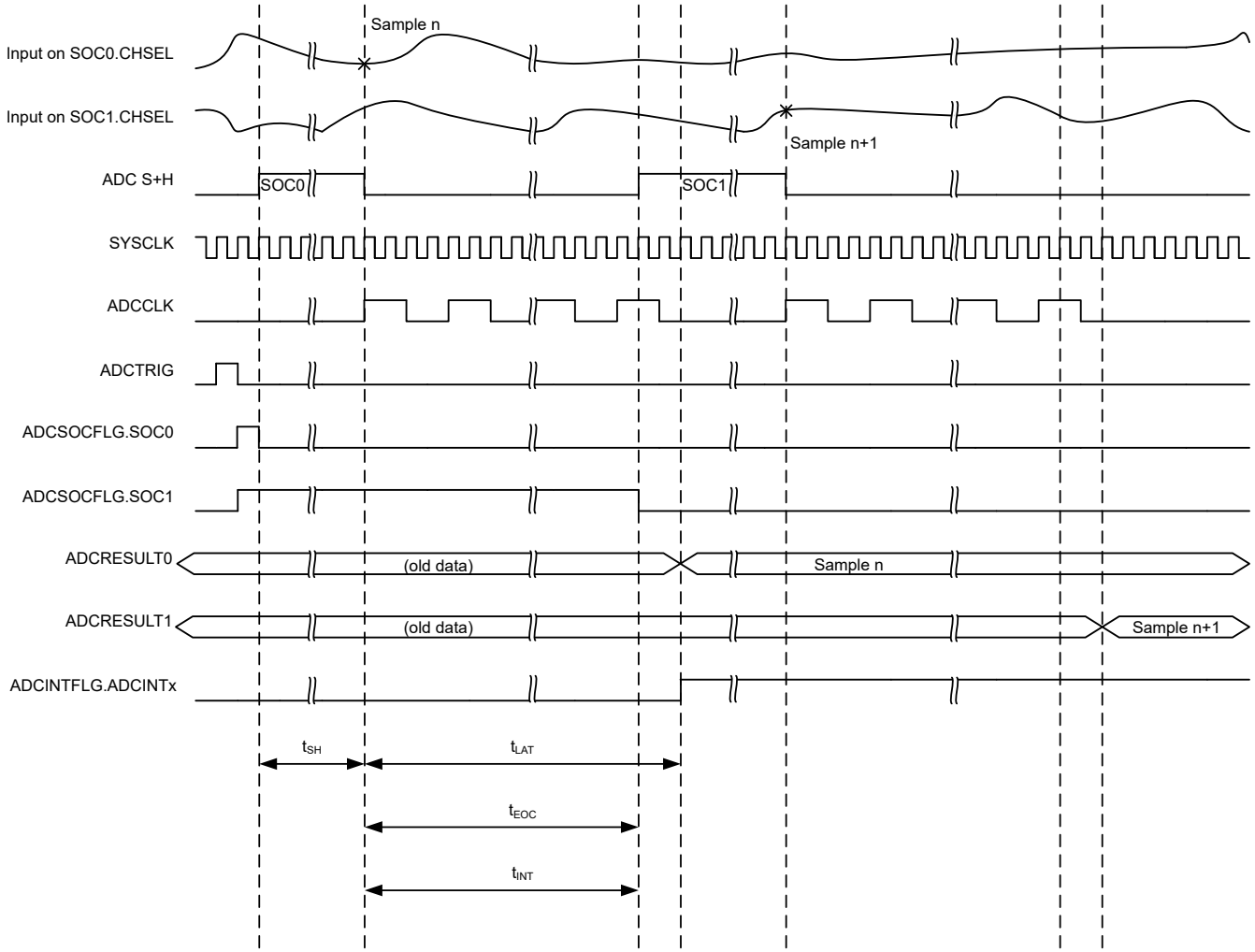


Figure 12-15. ADC Timings in Late Interrupt Mode

Table 12-10. ADC Timings with SAMPCAPRESETSEL = 0

ADCCLK Prescale		SYSCLK Cycles				
ADCCTL2. PRESCALE	Prescale Ratio	t_{EOC}	t_{LAT}	t_{INT} (Early)(1)	t_{INT} (Late)	t_{DMA}
0	1	12	16	1	12	16
2	2	24	28	1	24	28
4	3	36	40	1	36	40
6	4	48	52	1	48	52
8	5	60	64	1	60	64
10	6	72	76	1	72	76
12	7	84	88	1	84	88
14	8	96	100	1	96	100

(1) By default, t_{INT} occurs one SYSCLK cycle after the S+H window if INTPULSEPOS is 0. This can be changed by writing to the OFFSET field in the ADCINTCYCLE register.

Table 12-11. ADC Timings with SAMPCAPRESETSEL = 1

ADCCLK Prescale		SYSCLK Cycles				
ADCCTL2. PRESCALE	Prescale Ratio	t_{EOC}	t_{LAT}	t_{INT} (Early)(1)	t_{INT} (Late)	t_{DMA}
0	1	11	15	1	11	15
2	2	22	26	1	22	26
4	3	33	37	1	22	37
6	4	44	48	1	44	48
8	5	55	59	1	55	59
10	6	66	70	1	66	70
12	7	77	81	1	77	81
14	8	88	92	1	88	92

(1) By default, t_{INT} occurs one SYSCLK cycle after the S+H window if INTPULSEPOS is 0. This can be changed by writing to the OFFSET field in the ADCINTCYCLE register.

Note

It is recommended only to configure whole numbered integer prescalers in ADCCTL2.PRESCALE.

12.11.2 Post-Processing Block Timings

The value of ADCRESULT is always available at time t_{LAT}, as specified in Section 12.11.1. The value of ADCPPBxRESULT, and the limit and zero-crossing comparisons are available 1 SYSCLK cycle later, as long as multiple PPB instances do not point to the same SOC. Table 12-12 shows PPB result availability timings when there are no SOCs shared between multiple PPBs. In cases where multiple PPBs point to the same SOC, PPB results become available sequentially, starting with the lowest numbered PPB. The first ADCPPBxRESULT becomes available 1 SYSCLK cycle after t_{LAT}, and each subsequent ADCPPBxRESULT becomes available 1 SYSCLK cycle after the previous PPB has completed the limit, in-line, and zero-crossing comparison. The serialized PPB results are therefore spaced 2 or 3 SYSCLK cycles apart, depending on whether the comparison uses ADCPPBxRESULT or PSUM/SUM respectively. This timing is as shown in Table 12-13. PPB aggregation values (PSUM, SUM, PCOUNT, COUNT) are available 1 cycle after the associated ADCPPBxRESULT becomes available.

Table 12-12. PPB Result Timings (One PPB per SOC)

Register	Description	Results Available
ADCRESULTy	ADC result	t_{LAT}
ADCPPBxRESULT	PPB result	t_{LAT} + 1 SYSCLK
ADCPPBxPSUM	Oversampling partial sum	t_{LAT} + 2 SYSCLK
ADCPPBxSUM	Oversampling sum	t_{LAT} + 2 SYSCLK
ADCPPBxPCOUNT	Oversampling partial sample count	t_{LAT} + 2 SYSCLK
ADCPPBxCOUNT	Oversampling sample count	t_{LAT} + 2 SYSCLK
Comparison	Limit and zero-crossing comparison (if using PPBxRESULT)	t_{LAT} + 2 SYSCLK
Comparison	Limit and zero-crossing comparison (if using PSUM or SUM)	t_{LAT} + 3 SYSCLK

Table 12-13. PPB Result Timings (Multiple PPBs Configured to Same SOC)

Register	Description	Results Available
ADCRESULTy	ADC result	t_{LAT}
ADCPPBxRESULT	PPB result	Varies (PPBs process serially)
ADCPPBxPSUM	Oversampling partial sum	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxSUM	Oversampling sum	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxPSUM	Oversampling partial sum	ADCPPBxRESULT+ 1 SYSCLK
ADCPPBxSUM	Oversampling sum	ADCPPBxRESULT+ 1 SYSCLK
Comparison	Limit and zero-crossing comparison (if using PPBxRESULT)	t_{LAT} + 2 SYSCLK
Comparison	Limit and zero-crossing comparison (if using PSUM or SUM)	t_{LAT} + 3 SYSCLK

12.12 Additional Information

The following sections contain additional practical information.

12.12.1 Choosing an Acquisition Window Duration

For correct operation, the input signal to the ADC must be allowed adequate time to charge the sample and hold capacitor, Ch. Typically, the S+H duration is chosen such that the sampling capacitor is charged to within ½ LSB or ¼ LSB of the final value, depending on the tolerable settling error.

The best methodology to determine the required settling time is to simulate the ADC and ADC driving circuits to make sure adequate settling performance. See *ADC Input Circuit Evaluation for C2000 MCUs* and *Charge-Sharing Driving Circuits for C2000 ADCs* for additional guidance on ADC signal conditioning circuit design and evaluation.

An approximation of the required settling time can also be determined using an RC settling model. The time constant for the model is given by the equation:

$$\tau = (R_S + R_{on}) \times C_h + R_S \times \left(C_S + C_p\right) \quad (5)$$

And the number of time constants needed is given by the equation:

$$k = ln\left(\frac{2^n}{settling\ error}\right) - ln\left(\frac{C_S + C_P}{CH}\right) \quad (6)$$

So the total S+H time must be set to at least:

$$t = k \cdot \tau \quad (7)$$

Where the following parameters are provided by the ADC input model in the device data sheet:

- n = ADC resolution (in bits)
- R_{ON} = ADC sampling switch resistance (provided in Ω)
- C_H = ADC sampling capacitor (provided in pF)
- C_p = ADC channel parasitic input capacitance (provided in pF)

And the following parameters are dependent on the application design:

- settling error = tolerable settling error (in LSBs)
- R_s = ADC driving circuit source impedance (typically in Ω or kΩ)
- C_S = capacitance on ADC input pin (typically in pF or nF)

For example, assuming the following parameters:

- n = 12-bits
- R_{ON} = 500Ω
- C_H = 12.5pF
- C_p = 12.7pF
- settling error = ¼ LSB
- R_s = 180Ω
- C_s = 150pF

The time constant is calculated as:

$$\tau = (180\Omega + 500\Omega) \times 12.5pF + 180\Omega \times (150pF + 12.7pF) = 37.8ns \quad (8)$$

And the number of required time constants is:

$$k = ln\left(\frac{2^{12}}{0.25}\right) - ln\left(\frac{150pF + 12.7pF}{12.5pF}\right) = 9.70 - 2.57 = 7.13 \quad (9)$$

So the S+H time must be set to at least: 37.8ns × 7.13 = 270ns

If SYSCLK = 160MHz , then each SYSCLK cycle is 6.25ns. S+H duration is 270ns/6.25ns = 43.2 SYSCLK cycles, so ACQPS for this input is set to at least CEILING(43.2) – 1 = 42.

While this gives a rough estimate of the required acquisition window, a better method is to setup a circuit with the ADC input model, a model of the source impedance/capacitance, and any board parasitics in SPICE (or similar software) and simulate to verify that the sampling capacitor settles to the desired accuracy.

Note

The device data sheet specifies a minimum ADC S+H window duration. Do not use an ACQPS value that gives a duration less than this specification.

12.12.2 Result Register Mapping

The ADC results and the ADC PPB results are duplicated for each memory bus controller in the system. Bus controllers include all CPUs, DMAs, and CLAs present on the specific part family and part number. For each bus controller, no access configuration is needed to allow read access to the result registers, and no contention occurs in cases where multiple bus controllers try to read the ADC results simultaneously.

12.12.3 Internal Temperature Sensor

The internal temperature sensor measures the junction temperature of the device. The output of the sensor can be sampled with the ADC through an internal connection. This can be enabled on channel ADC22 on ADCA and on the CMPSS1_HP5 input by setting the ENABLE bit in the TSNSCTL register.

To convert the temperature sensor reading into a temperature, pass the temperature sensor reading to the ADC_getTemperatureC() function in .

12.12.4 Designing an External Reference Circuit

Figure 12-16 shows the basic organization of the external voltage reference generation circuitry. For best performance, the externally generated reference voltage must be buffered by a precision op-amp with good bandwidth and low output impedance before being driven into the ADC reference pin. A capacitor between the high and low reference pins must be placed on the PCB as close to the pins as practical to help absorb high-frequency currents. A series resistor (typically <1Ω) in series with this capacitor is sometimes necessary to maintain op-amp stability.

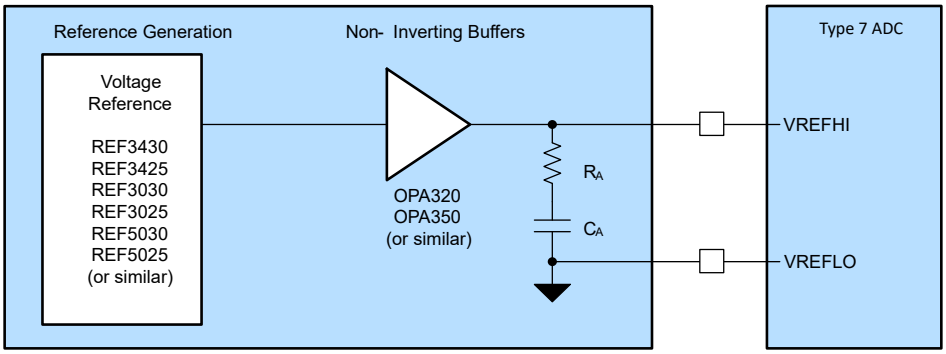


Figure 12-16. ADC Reference System

12.12.5 ADC-DAC Loopback Testing

For system diagnostic or functional safety purposes, the user application can perform a loopback test of the ADC module to verify that the ADC is converting correctly. Using the output of the DAC in the first CMPSS module, the device can be configured to supply a series of known voltages to the input of the converter, and the conversion result verified against expected results. Loopback test mode is enabled by setting the bit corresponding to the ADC module under test in the ADCLOOPBACK register in the analog subsystem module to 1. Figure 12-17 shows the connection between the CMPSS DAC output and the ADC.

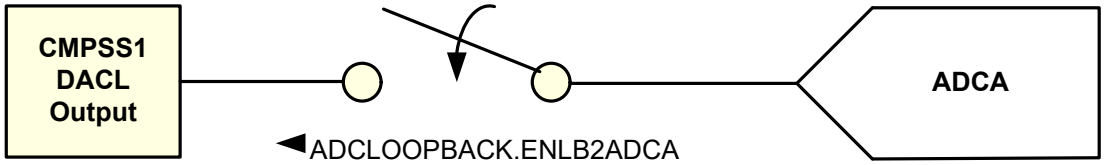


Figure 12-17. CMPSS to ADC Loopback Connection

In ADC loopback test mode, the following special considerations apply:

- The ADC module always samples the CMPSS1 DACL output, regardless of what channel is selected in the ADCSOCxCTL.CHSEL field.
- The minimum sampling window size (ACQPS) when converting the DAC output is 3.2µs (512 SYSCLK cycles at 160MHz SYSCLK).
- The output resolution of the CMPSS DAC is 6 bits. The lower 6 bits of the input DACVAL are discarded.
- ADC loopback test mode affects CMPSS trip voltages. Avoid enabling ADC loopback mode during regular CMPSS operation.

For more information on the CMPSS module and how to configure the CMPSS DAC, see the *Comparator Subsystem (CMPSS)* chapter.

12.12.6 Internal Test Mode

For diagnostic purposes, the ADC can sample various internal node voltages using a special input selection mux called INTERNALTEST. When internal test mode is enabled, the INTERNALTEST mux selection overrides the ADC-A input channel mux: ADC-A samples the INTERNALTEST selection instead of the channel selected by ADCSOCxCTL.CHSEL. Internal test mode can be used to sample the VDDCORE voltage, VREFLO, VDDA, VSSA, and the CMPSS DAC outputs.

To enable internal test mode, write the desired node selection to the TESTSEL field of the INTERNALTESTCTL analog subsystem register. For details of internal test mux connections to various internal device voltage nodes, refer to the INTERNALTESTCTL register description.

To disable internal test mode, write 0 to the TESTSEL field of the INTERNALTESTCTL register.

When using internal test mode, the following special considerations apply:
- INTERNALTESTCTL.TESTSEL overrides the value of ADCSOCxCTL.CHSEL on ADCA when a non-zero value is configured.
- The minimum sampling window size (ACQPS) when converting INTERNALTEST is 3.2µs (512 SYSCLK cycles at 160MHz SYSCLK).
- The effective resolution of the CMPSS DAC outputs to INTERNALTEST is 6 bits.

For more information on the CMPSS module and how to configure the CMPSS DAC, see the *Comparator Subsystem (CMPSS)* chapter.

12.13 Software

12.13.1 ADC Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/adc

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

12.13.1.1 ADC Software Triggering

FILE: adc_ex1_soc_software.c

This example converts some voltages on ADCA based on a software trigger.

External Connections
- A0, A4 should be connected to signals to convert

Watch Variables
- *myADC0Result0* - Digital representation of the voltage on pin A0
- *myADC0Result1* - Digital representation of the voltage on pin A4

12.13.1.2 ADC MCPWM Triggering

FILE: adc_ex2_soc_mcpwm.c

This example sets up MCPWM1 to periodically trigger a conversion on ADCA.

External Connections
- A0 should be connected to a signal to convert

Watch Variables
- *myADC0Results* - A sequence of analog-to-digital conversion samples from pin A0. The time between samples is determined based on the period of the MCPWM timer.

12.13.1.3 ADC Temperature Sensor Conversion

FILE: adc_ex3_temp_sensor.c

This example sets up the MCPWM to periodically trigger the ADC. The ADC converts the internal connection to the temperature sensor, which is then interpreted as a temperature by calling the ADC_getTemperatureC() function.

Watch Variables

- *sensorSample* - The raw reading from the temperature sensor
- *sensorTemp* - The interpretation of the sensor sample as a temperature in degrees Celsius.

12.13.1.4 ADC Continuous Conversions Read by DMA (adc_soc_continuous_dma)

FILE: adc_ex6_soc_continuous_dma.c

This example sets up ADC channel to conversion. The results will be transferred by the DMA into a buffer in RAM.

External Connections

- A3 pins should be connected to signals to convert

Watch Variables

- *myADC0DataBuffer* : a digital representation of the voltage on pin A3

12.13.1.5 ADC PPB Offset (adc_ppb_offset)

FILE: adc_ex7_ppb_offset.c

This example software triggers the ADC. Some SOCs have automatic offset adjustment applied by the post-processing block. After the program runs, the memory will contain ADC & post-processing block(PPB) results.

External Connections

- A5 pins should be connected to signals to convert

Watch Variables

- *myADC0Result* : a digital representation of the voltage on pin A5
- *myADC0PPBResult* : a digital representation of the voltage on pin A5, minus 30 LSBs of automatically added offset

12.13.1.6 ADC PPB Limits (adc_ppb_limits)

FILE: adc_ex8_ppb_limits.c

This example sets up the MCPWM to periodically trigger the ADC. If the results are outside of the defined range, the post-processing block will generate an interrupt.

The default limits are 1000LSBs and 3000LSBs. With VREFHI set to 3.3V, the PPB will generate an interrupt if the input voltage goes above about 2.4V or below about 0.8V.

External Connections

- A0 should be connected to a signal to convert

Watch Variables

- None

12.13.1.7 ADC SOC Oversampling

FILE: adc_ex13_soc_oversampling.c

This example sets up MCPWM1 to periodically trigger a set of conversions on ADCA including multiple SOCs that all convert A2 to achieve oversampling on A2.

ADCA Interrupt ISRs are used to read results of ADCA.

External Connections

- A0, A5, A8 should be connected to signals to be converted.

Watch Variables

- *adcAResult0* - Digital representation of the voltage on pin A0
- *adcAResult1* - Digital representation of the voltage on pin A5
- *adcAResult2* - Digital representation of the voltage on pin A8

12.13.1.8 ADC Trigger Repeater Oversampling

FILE: adc_ex15_trigger_repeater_oversampling.c

This example configures ADC for oversampling using trigger repeater block. The MCPWM1 is configured to periodically trigger the ADC SOC and the trigger repeater module is configured to generate 4 repeated pulses. Post-processing block will take the repeated pulses, accumulates them and stores the results in ppb sum register.

External Connections

- A0 should be connected to signals to convert.

Watch Variables

- *myADC0Result* - Digital representation of the voltage on pin A0
- *myPPB0Result* - Digital representation of the voltage of the 4 repeated pulses on pin A0

12.14 ADC Registers

This Section describes the ADC Registers.

12.14.1 ADC Base Address Table

Table 12-14. ADC Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
AdcaResultRegs	ADC_LITE_RESULT_REGS	ADCARESULT_BASE	0x0000_1800	YES	-
AdcaRegs	ADC_LITE_REGS	ADCA_BASE	0x0000_7400	-	YES

12.14.2 ADC_LITE_RESULT_REGS Registers

Table 12-15 lists the memory-mapped registers for the ADC_LITE_RESULT_REGS registers. All register offset addresses not listed in Table 12-15 should be considered as reserved locations and the register contents should not be modified.

Table 12-15. ADC_LITE_RESULT_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	ADCRESULT0	ADC Result 0 Register	
1h	ADCRESULT1	ADC Result 1 Register	
2h	ADCRESULT2	ADC Result 2 Register	
3h	ADCRESULT3	ADC Result 3 Register	
4h	ADCRESULT4	ADC Result 4 Register	
5h	ADCRESULT5	ADC Result 5 Register	
6h	ADCRESULT6	ADC Result 6 Register	
7h	ADCRESULT7	ADC Result 7 Register	
8h	ADCRESULT8	ADC Result 8 Register	
9h	ADCRESULT9	ADC Result 9 Register	
Ah	ADCRESULT10	ADC Result 10 Register	
Bh	ADCRESULT11	ADC Result 11 Register	
Ch	ADCRESULT12	ADC Result 12 Register	
Dh	ADCRESULT13	ADC Result 13 Register	
Eh	ADCRESULT14	ADC Result 14 Register	
Fh	ADCRESULT15	ADC Result 15 Register	
20h	ADCPPB1RESULT	ADC Post Processing Block 1 Result Register	
22h	ADCPPB2RESULT	ADC Post Processing Block 2 Result Register	
24h	ADCPPB3RESULT	ADC Post Processing Block 3 Result Register	
28h	ADCPPB1SUM	ADC PPB 1 Final Sum Result Register	
2Ah	ADCPPB1COUNT	ADC PPB1 Final Conversion Count Register	

Complex bit access types are encoded to fit into small table cells. Table 12-16 shows the codes that are used for access types in this section.

Table 12-16. ADC_LITE_RESULT_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

12.14.2.1 ADCRESULT0 Register (Offset = 0h) [Reset = 0000h]

ADCRESULT0 is shown in Figure 12-18 and described in Table 12-17.

Return to the Summary Table.

ADC Result 0 Register

Figure 12-18. ADCRESULT0 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-17. ADCRESULT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 0 12-bit ADC result. After the ADC completes a conversion of SOC0, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.2 ADCRESULT1 Register (Offset = 1h) [Reset = 0000h]

ADCRESULT1 is shown in Figure 12-19 and described in Table 12-18.

Return to the Summary Table.

ADC Result 1 Register

Figure 12-19. ADCRESULT1 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-18. ADCRESULT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 1 12-bit ADC result. After the ADC completes a conversion of SOC1, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.3 ADCRESULT2 Register (Offset = 2h) [Reset = 0000h]

ADCRESULT2 is shown in Figure 12-20 and described in Table 12-19.

Return to the Summary Table.

ADC Result 2 Register

Figure 12-20. ADCRESULT2 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-19. ADCRESULT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 2 12-bit ADC result. After the ADC completes a conversion of SOC2, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.4 ADCRESULT3 Register (Offset = 3h) [Reset = 0000h]

ADCRESULT3 is shown in Figure 12-21 and described in Table 12-20.

Return to the Summary Table.

ADC Result 3 Register

Figure 12-21. ADCRESULT3 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-20. ADCRESULT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 3 12-bit ADC result. After the ADC completes a conversion of SOC3, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.5 ADCRESULT4 Register (Offset = 4h) [Reset = 0000h]

ADCRESULT4 is shown in Figure 12-22 and described in Table 12-21.

Return to the Summary Table.

ADC Result 4 Register

Figure 12-22. ADCRESULT4 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-21. ADCRESULT4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 4 12-bit ADC result. After the ADC completes a conversion of SOC4, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.6 ADCRESULT5 Register (Offset = 5h) [Reset = 0000h]

ADCRESULT5 is shown in Figure 12-23 and described in Table 12-22.

Return to the Summary Table.

ADC Result 5 Register

Figure 12-23. ADCRESULT5 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-22. ADCRESULT5 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 5 12-bit ADC result. After the ADC completes a conversion of SOC5, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.7 ADCRESULT6 Register (Offset = 6h) [Reset = 0000h]

ADCRESULT6 is shown in Figure 12-24 and described in Table 12-23.

Return to the Summary Table.

ADC Result 6 Register

Figure 12-24. ADCRESULT6 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-23. ADCRESULT6 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 6 12-bit ADC result. After the ADC completes a conversion of SOC6, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.8 ADCRESULT7 Register (Offset = 7h) [Reset = 0000h]

ADCRESULT7 is shown in Figure 12-25 and described in Table 12-24.

Return to the Summary Table.

ADC Result 7 Register

Figure 12-25. ADCRESULT7 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-24. ADCRESULT7 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 7 12-bit ADC result. After the ADC completes a conversion of SOC7, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.9 ADCRESULT8 Register (Offset = 8h) [Reset = 0000h]

ADCRESULT8 is shown in Figure 12-26 and described in Table 12-25.

Return to the Summary Table.

ADC Result 8 Register

Figure 12-26. ADCRESULT8 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-25. ADCRESULT8 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 8 12-bit ADC result. After the ADC completes a conversion of SOC8, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.10 ADCRESULT9 Register (Offset = 9h) [Reset = 0000h]

ADCRESULT9 is shown in Figure 12-27 and described in Table 12-26.

Return to the Summary Table.

ADC Result 9 Register

Figure 12-27. ADCRESULT9 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-26. ADCRESULT9 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 9 12-bit ADC result. After the ADC completes a conversion of SOC9, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.11 ADCRESULT10 Register (Offset = Ah) [Reset = 0000h]

ADCRESULT10 is shown in Figure 12-28 and described in Table 12-27.

Return to the Summary Table.

ADC Result 10 Register

Figure 12-28. ADCRESULT10 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-27. ADCRESULT10 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 10 12-bit ADC result. After the ADC completes a conversion of SOC10, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.12 ADCRESULT11 Register (Offset = Bh) [Reset = 0000h]

ADCRESULT11 is shown in Figure 12-29 and described in Table 12-28.

Return to the Summary Table.

ADC Result 11 Register

Figure 12-29. ADCRESULT11 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-28. ADCRESULT11 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 11 12-bit ADC result. After the ADC completes a conversion of SOC11, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.13 ADCRESULT12 Register (Offset = Ch) [Reset = 0000h]

ADCRESULT12 is shown in Figure 12-30 and described in Table 12-29.

Return to the Summary Table.

ADC Result 12 Register

Figure 12-30. ADCRESULT12 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-29. ADCRESULT12 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 12 12-bit ADC result. After the ADC completes a conversion of SOC12, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.14 ADCRESULT13 Register (Offset = Dh) [Reset = 0000h]

ADCRESULT13 is shown in Figure 12-31 and described in Table 12-30.

Return to the Summary Table.

ADC Result 13 Register

Figure 12-31. ADCRESULT13 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-30. ADCRESULT13 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 13 12-bit ADC result. After the ADC completes a conversion of SOC13, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.15 ADCRESULT14 Register (Offset = Eh) [Reset = 0000h]

ADCRESULT14 is shown in Figure 12-32 and described in Table 12-31.

Return to the Summary Table.

ADC Result 14 Register

Figure 12-32. ADCRESULT14 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-31. ADCRESULT14 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 14 12-bit ADC result. After the ADC completes a conversion of SOC14, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.16 ADCRESULT15 Register (Offset = Fh) [Reset = 0000h]

ADCRESULT15 is shown in Figure 12-33 and described in Table 12-32.

Return to the Summary Table.

ADC Result 15 Register

Figure 12-33. ADCRESULT15 Register

15	14	13	12	11	10	9	8
RESERVED				RESULT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 12-32. ADCRESULT15 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	RESULT	R	0h	ADC Result 15 12-bit ADC result. After the ADC completes a conversion of SOC15, the digital result is placed in this bit field. Reset type: SYSRSn

12.14.2.17 ADCPPB1RESULT Register (Offset = 20h) [Reset = 00000000h]

ADCPPB1RESULT is shown in Figure 12-34 and described in Table 12-33.

Return to the Summary Table.

ADC Post Processing Block 1 Result Register

Figure 12-34. ADCPPB1RESULT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGN																			PPBRESULT												
R-0h																			R-0h											

Table 12-33. ADCPPB1RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 1 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

12.14.2.18 ADCPPB2RESULT Register (Offset = 22h) [Reset = 00000000h]

ADCPPB2RESULT is shown in Figure 12-35 and described in Table 12-34.

Return to the Summary Table.

ADC Post Processing Block 2 Result Register

Figure 12-35. ADCPPB2RESULT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGN																			PPBRESULT												
R-0h																			R-0h											

Table 12-34. ADCPPB2RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 2 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

12.14.2.19 ADCPPB3RESULT Register (Offset = 24h) [Reset = 00000000h]

ADCPPB3RESULT is shown in Figure 12-36 and described in Table 12-35.

Return to the Summary Table.

ADC Post Processing Block 3 Result Register

Figure 12-36. ADCPPB3RESULT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGN																			PPBRESULT												
R-0h																			R-0h											

Table 12-35. ADCPPB3RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 3 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

12.14.2.20 ADCPPB1SUM Register (Offset = 28h) [Reset = 00000000h]

ADCPPB1SUM is shown in Figure 12-37 and described in Table 12-36.

Return to the Summary Table.

ADC PPB 1 Final Sum Result Register

Figure 12-37. ADCPPB1SUM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGN																SUM															
R-0h																R-0h															

Table 12-36. ADCPPB1SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	SUM	R	0h	Post Processing Block 1 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

12.14.2.21 ADCPPB1COUNT Register (Offset = 2Ah) [Reset = 0000h]

ADCPPB1COUNT is shown in Figure 12-38 and described in Table 12-37.

Return to the Summary Table.

ADC PPB1 Final Conversion Count Register

Figure 12-38. ADCPPB1COUNT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				COUNT			
R-0h				R-0h			

Table 12-37. ADCPPB1COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	COUNT	R	0h	Post Processing Block 1 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

12.14.3 ADC_LITE_REGS Registers

Table 12-38 lists the memory-mapped registers for the ADC_LITE_REGS registers. All register offset addresses not listed in Table 12-38 should be considered as reserved locations and the register contents should not be modified.

Table 12-38. ADC_LITE_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	ADCCTL1	ADC Control 1 Register	EALLOW
2h	ADCCTL2	ADC Control 2 Register	EALLOW
8h	ADCINTSEL	ADC Interrupt 1, 2, 3 and 4 Selection Register	EALLOW
Ah	ADCDMAINTSEL	ADC DMA Interrupt 1, 2, 3 and 4 Selection Register	EALLOW
Ch	ADCRAWINTFLG	ADC Raw Interrupt Flag Register	
Eh	ADCINTFLG	ADC Interrupt Flag Register	
10h	ADCINTFLGFRC	ADC Interrupt Flag Force Register	
12h	ADCINTFLGCLR	ADC Interrupt Flag Clear Register	
14h	ADCINTOVF	ADC Interrupt Overflow Register	
16h	ADCINTOVFCLR	ADC Interrupt Overflow Clear Register	
18h	ADCSOCPRICTL	ADC SOC Priority Control Register	EALLOW
1Ah	ADCINTSOCSEL1	ADC Interrupt SOC Selection 1 Register	EALLOW
1Eh	ADCSOCFLG1	ADC SOC Flag 1 Register	
20h	ADCSOCFRC1	ADC SOC Force 1 Register	
22h	ADCSOCOVF1	ADC SOC Overflow 1 Register	
24h	ADCSOCOVFCLR1	ADC SOC Overflow Clear 1 Register	
26h	ADCSOC0CTL	ADC SOC0 Control Register	EALLOW
28h	ADCSOC1CTL	ADC SOC1 Control Register	EALLOW
2Ah	ADCSOC2CTL	ADC SOC2 Control Register	EALLOW
2Ch	ADCSOC3CTL	ADC SOC3 Control Register	EALLOW
2Eh	ADCSOC4CTL	ADC SOC4 Control Register	EALLOW
30h	ADCSOC5CTL	ADC SOC5 Control Register	EALLOW
32h	ADCSOC6CTL	ADC SOC6 Control Register	EALLOW
34h	ADCSOC7CTL	ADC SOC7 Control Register	EALLOW
36h	ADCSOC8CTL	ADC SOC8 Control Register	EALLOW
38h	ADCSOC9CTL	ADC SOC9 Control Register	EALLOW
3Ah	ADCSOC10CTL	ADC SOC10 Control Register	EALLOW
3Ch	ADCSOC11CTL	ADC SOC11 Control Register	EALLOW
3Eh	ADCSOC12CTL	ADC SOC12 Control Register	EALLOW
40h	ADCSOC13CTL	ADC SOC13 Control Register	EALLOW
42h	ADCSOC14CTL	ADC SOC14 Control Register	EALLOW
44h	ADCSOC15CTL	ADC SOC15 Control Register	EALLOW
66h	ADCEVTSTAT	ADC Event Status Register	
68h	ADCEVTCLR	ADC Event Clear Register	
6Ah	ADCEVTSEL	ADC Event Selection Register	EALLOW
6Ch	ADCEVTINTSEL	ADC Event Interrupt Selection Register	EALLOW
72h	ADCREV	ADC Revision Register	
74h	ADCOFFTRIM	ADC Offset Trim Register 1	EALLOW
80h	ADCPPB1CONFIG	ADC PPB{#} Config Register	EALLOW
84h	ADCPPB1OFFCAL	ADC PPB1 Offset Calibration Register	EALLOW

Table 12-38. ADC_LITE_REGS Registers (continued)

Offset	Acronym	Register Name	Write Protection
86h	ADCPPB1OFFREF	ADC PPB1 Offset Reference Register	
88h	ADCPPB1TRIPHI	ADC PPB1 Trip High Register	EALLOW
8Ah	ADCPPB1TRIPLO	ADC PPB1 Trip Low/Trigger Time Stamp Register	EALLOW
90h	ADCPPB2CONFIG	ADC PPB{#} Config Register	EALLOW
94h	ADCPPB2OFFCAL	ADC PPB2 Offset Calibration Register	EALLOW
96h	ADCPPB2OFFREF	ADC PPB2 Offset Reference Register	
98h	ADCPPB2TRIPHI	ADC PPB2 Trip High Register	EALLOW
9Ah	ADCPPB2TRIPLO	ADC PPB2 Trip Low/Trigger Time Stamp Register	EALLOW
A0h	ADCPPB3CONFIG	ADC PPB{#} Config Register	EALLOW
A4h	ADCPPB3OFFCAL	ADC PPB3 Offset Calibration Register	EALLOW
A6h	ADCPPB3OFFREF	ADC PPB3 Offset Reference Register	
A8h	ADCPPB3TRIPHI	ADC PPB3 Trip High Register	EALLOW
AAh	ADCPPB3TRIPLO	ADC PPB3 Trip Low/Trigger Time Stamp Register	EALLOW
C0h	ADCINTCYCLE	ADC Early Interrupt Generation Cycle	EALLOW
C2h	ADCINLTRIM1	ADC Linearity Trim 1 Register	EALLOW
C4h	ADCINLTRIM2	ADC Linearity Trim 2 Register	EALLOW
CEh	ADCREV2	ADC Wrapper Revision Register	
E0h	REP1CTL	ADC Trigger Repeater 1 Control Register	EALLOW
E2h	REP1N	ADC Trigger Repeater 1 N Select Register	EALLOW
E6h	REP1SPREAD	ADC Trigger Repeater 1 Spread Select Register	EALLOW
E8h	REP1FRC	ADC Trigger Repeater 1 Software Force Register	EALLOW
100h	ADCPPB1LIMIT	ADC PPB1Conversion Count Limit Register	EALLOW
102h	ADCPPBP1PCOUNT	ADC PPB1 Partial Conversion Count Register	
104h	ADCPPB1CONFIG2	ADC PPB1 Sum Shift Register	
106h	ADCPPB1PSUM	ADC PPB1 Partial Sum Register	

Complex bit access types are encoded to fit into small table cells. Table 12-39 shows the codes that are used for access types in this section.

Table 12-39. ADC_LITE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 12-39. ADC_LITE_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

12.14.3.1 ADCCTL1 Register (Offset = 0h) [Reset = 00000000h]

ADCCTL1 is shown in Figure 12-39 and described in Table 12-40.

Return to the Summary Table.

ADC Control 1 Register

Figure 12-39. ADCCTL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		ADCBSY	RESERVED	ADCBSYCHN			
R-0h		R-0h	R-0h	R-0h			
7	6	5	4	3	2	1	0
ADCPWDNZ	RESERVED				INTPULSEPOS	RESERVED	
R/W-0h	R-0h				R/W-0h	R-0h	

Table 12-40. ADCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	Reserved
13	ADCBSY	R	0h	ADC Busy. Set when ADC SOC is generated, cleared by hardware four ADC clocks after negative edge of S/H pulse. Used by the ADC state machine to determine if ADC is available to sample. 0 ADC is available to sample next channel 1 ADC is busy and cannot sample another channel Reset type: SYSRSn
12	RESERVED	R	0h	Reserved
11-8	ADCBSYCHN	R	0h	ADC Busy Channel. Set when an ADC Start of Conversion (SOC) is generated. When ADCBSY=0: holds the value of the last converted SOC When ADCBSY=1: reflects the SOC currently being processed 0h SOC0 is currently processing or was last SOC converted 1h SOC1 is currently processing or was last SOC converted 2h SOC2 is currently processing or was last SOC converted 3h SOC3 is currently processing or was last SOC converted 4h SOC4 is currently processing or was last SOC converted 5h SOC5 is currently processing or was last SOC converted 6h SOC6 is currently processing or was last SOC converted 7h SOC7 is currently processing or was last SOC converted 8h SOC8 is currently processing or was last SOC converted 9h SOC9 is currently processing or was last SOC converted Ah SOC10 is currently processing or was last SOC converted Bh SOC11 is currently processing or was last SOC converted Ch SOC12 is currently processing or was last SOC converted Dh SOC13 is currently processing or was last SOC converted Eh SOC14 is currently processing or was last SOC converted Fh SOC15 is currently processing or was last SOC converted Reset type: SYSRSn
7	ADCPWDNZ	R/W	0h	ADC Power Down (active low). This bit controls the power up and power down of all the analog circuitry inside the analog core. 0 All analog circuitry inside the core is powered down 1 All analog circuitry inside the core is powered up Reset type: SYSRSn

Table 12-40. ADCCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-3	RESERVED	R	0h	Reserved
2	INTPULSEPOS	R/W	0h	ADC Interrupt Pulse Position. 0 Interrupt pulse generation occurs when ADC begins conversion (at the end of the acquisition window) plus a number of SYSCLK cycles as specified in the ADCINTCYCLE.OFFSET register. 1 Interrupt pulse generation occurs at the end of the conversion, 1 cycle prior to the ADC result latching into its result register Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

12.14.3.2 ADCCTL2 Register (Offset = 2h) [Reset = 00000000h]

ADCCTL2 is shown in Figure 12-40 and described in Table 12-41.

Return to the Summary Table.

ADC Control 2 Register

Figure 12-40. ADCCTL2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED			RESERVED				RESERVED
R-0h			R-0h				R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED		PRESCALE			
R/W-0h	R/W-0h	R-0h		R/W-0h			

Table 12-41. ADCCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12-9	RESERVED	R	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R	0h	Reserved
3-0	PRESCALE	R/W	0h	ADC Clock Prescaler. 0000 ADCCLK = Input Clock / 1.0 0001 ADCCLK = Input Clock / 1.5 0010 ADCCLK = Input Clock / 2.0 0011 ADCCLK = Input Clock / 2.5 0100 ADCCLK = Input Clock / 3.0 0101 ADCCLK = Input Clock / 3.5 0110 ADCCLK = Input Clock / 4.0 0111 ADCCLK = Input Clock / 4.5 1000 ADCCLK = Input Clock / 5.0 1001 ADCCLK = Input Clock / 5.5 1010 ADCCLK = Input Clock / 6.0 1011 ADCCLK = Input Clock / 6.5 1100 ADCCLK = Input Clock / 7.0 1101 ADCCLK = Input Clock / 7.5 1110 ADCCLK = Input Clock / 8.0 1111 ADCCLK = Input Clock / 8.5 Note: Non-integer ADC clock dividers are not recommended. Reset type: SYSRSn

12.14.3.3 ADCINTSEL Register (Offset = 8h) [Reset = 00000000h]

ADCINTSEL is shown in Figure 12-41 and described in Table 12-42.

Return to the Summary Table.

ADC Interrupt 1, 2, 3 and 4 Selection Register

Figure 12-41. ADCINTSEL Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED					
R/W-0h	R/W-0h	R/W-0h					
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED					
R/W-0h	R/W-0h	R/W-0h					
15	14	13	12	11	10	9	8
INT2E	INT2CONT	INT2SEL					
R/W-0h	R/W-0h	R/W-0h					
7	6	5	4	3	2	1	0
INT1E	INT1CONT	INT1SEL					
R/W-0h	R/W-0h	R/W-0h					

Table 12-42. ADCINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29-24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21-16	RESERVED	R/W	0h	Reserved
15	INT2E	R/W	0h	ADCINT2 Interrupt Enable 0 ADCINT2 is disabled 1 ADCINT2 is enabled Reset type: SYSRSn
14	INT2CONT	R/W	0h	ADCINT2 Continue to Interrupt Mode 0 No further ADCINT2 pulses are generated until ADCINT2 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT2 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn

Table 12-42. ADCINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-8	INT2SEL	R/W	0h	ADCINT2 EOC Source Select 00h EOC0 is trigger for ADCINT2 01h EOC1 is trigger for ADCINT2 02h EOC2 is trigger for ADCINT2 03h EOC3 is trigger for ADCINT2 04h EOC4 is trigger for ADCINT2 05h EOC5 is trigger for ADCINT2 06h EOC6 is trigger for ADCINT2 07h EOC7 is trigger for ADCINT2 08h EOC8 is trigger for ADCINT2 09h EOC9 is trigger for ADCINT2 0Ah EOC10 is trigger for ADCINT2 0Bh EOC11 is trigger for ADCINT2 0Ch EOC12 is trigger for ADCINT2 0Dh EOC13 is trigger for ADCINT2 0Eh EOC14 is trigger for ADCINT2 0Fh EOC15 is trigger for ADCINT2 10h - 1Fh Reserved 20h OSINT1 is trigger for ADCINT2 21h - 3Fh Reserved Reset type: SYSRSn
7	INT1E	R/W	0h	ADCINT1 Interrupt Enable 0 ADCINT1 is disabled 1 ADCINT1 is enabled Reset type: SYSRSn
6	INT1CONT	R/W	0h	ADCINT1 Continue to Interrupt Mode 0 No further ADCINT1 pulses are generated until ADCINT1 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT1 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
5-0	INT1SEL	R/W	0h	ADCINT1 EOC Source Select 00h EOC0 is trigger for ADCINT1 01h EOC1 is trigger for ADCINT1 02h EOC2 is trigger for ADCINT1 03h EOC3 is trigger for ADCINT1 04h EOC4 is trigger for ADCINT1 05h EOC5 is trigger for ADCINT1 06h EOC6 is trigger for ADCINT1 07h EOC7 is trigger for ADCINT1 08h EOC8 is trigger for ADCINT1 09h EOC9 is trigger for ADCINT1 0Ah EOC10 is trigger for ADCINT1 0Bh EOC11 is trigger for ADCINT1 0Ch EOC12 is trigger for ADCINT1 0Dh EOC13 is trigger for ADCINT1 0Eh EOC14 is trigger for ADCINT1 0Fh EOC15 is trigger for ADCINT1 10h - 1Fh Reserved 20h OSINT1 is trigger for ADCINT1 21h - 3Fh Reserved Reset type: SYSRSn

12.14.3.4 ADCDMAINTSEL Register (Offset = Ah) [Reset = 00000000h]

ADCDMAINTSEL is shown in Figure 12-42 and described in Table 12-43.

Return to the Summary Table.

ADC DMA Interrupt 1, 2, 3 and 4 Selection Register

Figure 12-42. ADCDMAINTSEL Register

31	30	29	28	27	26	25	24
RESERVED	RESERVED	RESERVED					
R/W-0h	R/W-0h	R/W-0h					
23	22	21	20	19	18	17	16
RESERVED	RESERVED	RESERVED					
R/W-0h	R/W-0h	R/W-0h					
15	14	13	12	11	10	9	8
DMAINT2E	DMAINT2CONT	DMAINT2SEL					
R/W-0h	R/W-0h	R/W-0h					
7	6	5	4	3	2	1	0
DMAINT1E	DMAINT1CONT	DMAINT1SEL					
R/W-0h	R/W-0h	R/W-0h					

Table 12-43. ADCDMAINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29-24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21-16	RESERVED	R/W	0h	Reserved
15	DMAINT2E	R/W	0h	ADCDMAINT2 Interrupt Enable 0 ADCDMAINT2 is disabled 1 ADCDMAINT2 is enabled Reset type: SYSRSn
14	DMAINT2CONT	R/W	0h	ADCDMAINT2 Continue to Interrupt Mode 0 No further ADCDMAINT2 pulses are generated until ADCDMAINT2 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT2 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn

Table 12-43. ADCDMAINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-8	DMAINT2SEL	R/W	0h	ADCDMAINT2 EOC Source Select 00h EOC0 is trigger for ADCDMAINT2 01h EOC1 is trigger for ADCDMAINT2 02h EOC2 is trigger for ADCDMAINT2 03h EOC3 is trigger for ADCDMAINT2 04h EOC4 is trigger for ADCDMAINT2 05h EOC5 is trigger for ADCDMAINT2 06h EOC6 is trigger for ADCDMAINT2 07h EOC7 is trigger for ADCDMAINT2 08h EOC8 is trigger for ADCDMAINT2 09h EOC9 is trigger for ADCDMAINT2 0Ah EOC10 is trigger for ADCDMAINT2 0Bh EOC11 is trigger for ADCDMAINT2 0Ch EOC12 is trigger for ADCDMAINT2 0Dh EOC13 is trigger for ADCDMAINT2 0Eh EOC14 is trigger for ADCDMAINT2 0Fh EOC15 is trigger for ADCDMAINT2 10h - 1Fh Reserved 20h OSINT1 is trigger for ADCDMAINT2 21h - 3Fh Reserved Reset type: SYSRSn
7	DMAINT1E	R/W	0h	ADCDMAINT1 Interrupt Enable 0 ADCDMAINT1 is disabled 1 ADCDMAINT1 is enabled Reset type: SYSRSn
6	DMAINT1CONT	R/W	0h	ADCDMAINT1 Continue to Interrupt Mode 0 No further ADCDMAINT1 pulses are generated until ADCDMAINT1 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT1 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
5-0	DMAINT1SEL	R/W	0h	ADCDMAINT1 EOC Source Select 00h EOC0 is trigger for ADCDMAINT1 01h EOC1 is trigger for ADCDMAINT1 02h EOC2 is trigger for ADCDMAINT1 03h EOC3 is trigger for ADCDMAINT1 04h EOC4 is trigger for ADCDMAINT1 05h EOC5 is trigger for ADCDMAINT1 06h EOC6 is trigger for ADCDMAINT1 07h EOC7 is trigger for ADCDMAINT1 08h EOC8 is trigger for ADCDMAINT1 09h EOC9 is trigger for ADCDMAINT1 0Ah EOC10 is trigger for ADCDMAINT1 0Bh EOC11 is trigger for ADCDMAINT1 0Ch EOC12 is trigger for ADCDMAINT1 0Dh EOC13 is trigger for ADCDMAINT1 0Eh EOC14 is trigger for ADCDMAINT1 0Fh EOC15 is trigger for ADCDMAINT1 10h - 1Fh Reserved 20h OSINT1 is trigger for ADCDMAINT1 21h - 3Fh Reserved Reset type: SYSRSn

12.14.3.5 ADCRAWINTFLG Register (Offset = Ch) [Reset = 00000000h]

ADCRAWINTFLG is shown in Figure 12-43 and described in Table 12-44.

Return to the Summary Table.

ADC Raw Interrupt Flag Register

Figure 12-43. ADCRAWINTFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	ADCDMARAWINT2	ADCDMARAWINT1
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	ADCRAWINT2	ADCRAWINT1
R-0h				R-0h	R-0h	R-0h	R-0h

Table 12-44. ADCRAWINTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	ADCDMARAWINT2	R	0h	ADC DMA Raw Interrupt 2 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
16	ADCDMARAWINT1	R	0h	ADC DMA Raw Interrupt 1 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	ADCRAWINT2	R	0h	ADC RAW Interrupt 2 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occured Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn

Table 12-44. ADCRAWINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ADCRAWINT1	R	0h	ADC RAW Interrupt 1 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn

12.14.3.6 ADCINTFLG Register (Offset = Eh) [Reset = 00000000h]

ADCINTFLG is shown in Figure 12-44 and described in Table 12-45.

Return to the Summary Table.

ADC Interrupt Flag Register

Figure 12-44. ADCINTFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	ADCDMAINT2	ADCDMAINT1
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED				RESERVED	RESERVED	ADCINT2RESULT	ADCINT1RESULT
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	ADCINT2	ADCINT1
R-0h				R-0h	R-0h	R-0h	R-0h

Table 12-45. ADCINTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	ADCDMAINT2	R	0h	ADC DMA Interrupt 2 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interrupt pulse generated 1 ADC DMA interrupt pulse generated If the ADC DMA interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
16	ADCDMAINT1	R	0h	ADC DMA Interrupt 1 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interrupt pulse generated 1 ADC DMA interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved

Table 12-45. ADCINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	ADCINT2RESULT	R	0h	ADC Interrupt 2 Results Ready Flag. This flag is set when the conversions results associated with ADCINT2 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT2 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
8	ADCINT1RESULT	R	0h	ADC Interrupt 1 Results Ready Flag. This flag is set when the conversions results associated with ADCINT1 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT1 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	ADCINT2	R	0h	ADC Interrupt 2 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
0	ADCINT1	R	0h	ADC Interrupt 1 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn

12.14.3.7 ADCINTFLGFRC Register (Offset = 10h) [Reset = 00000000h]

ADCINTFLGFRC is shown in Figure 12-45 and described in Table 12-46.

Return to the Summary Table.

ADC Interrupt Flag Force Register

Figure 12-45. ADCINTFLGFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	ADCDMAINT2	ADCDMAINT1
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	ADCINT2	ADCINT1
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 12-46. ADCINTFLGFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	RESERVED	R-0/W1S	0h	Reserved
18	RESERVED	R-0/W1S	0h	Reserved
17	ADCDMAINT2	R-0/W1S	0h	ADC DMA interrupt 2 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT2 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
16	ADCDMAINT1	R-0/W1S	0h	ADC DMA interrupt 1 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT1 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	RESERVED	R-0/W1S	0h	Reserved
2	RESERVED	R-0/W1S	0h	Reserved
1	ADCINT2	R-0/W1S	0h	ADC Interrupt 2 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT2 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
0	ADCINT1	R-0/W1S	0h	ADC Interrupt 1 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT1 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn

12.14.3.8 ADCINTFLGCLR Register (Offset = 12h) [Reset = 00000000h]

ADCINTFLGCLR is shown in Figure 12-46 and described in Table 12-47.

Return to the Summary Table.

ADC Interrupt Flag Clear Register

Figure 12-46. ADCINTFLGCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	ADCDMAINT2	ADCDMAINT1
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	ADCINT2	ADCINT1
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 12-47. ADCINTFLGCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	RESERVED	R-0/W1C	0h	Reserved
18	RESERVED	R-0/W1C	0h	Reserved
17	ADCDMAINT2	R-0/W1C	0h	ADC DMA Interrupt 2 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT2 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
16	ADCDMAINT1	R-0/W1C	0h	ADC DMA Interrupt 1 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT1 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	RESERVED	R-0/W1C	0h	Reserved
2	RESERVED	R-0/W1C	0h	Reserved
1	ADCINT2	R-0/W1C	0h	ADC Interrupt 2 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT2 and ADCINT2RESULT flags in the ADCINTFLG, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn

Table 12-47. ADCINTFLGCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ADCINT1	R-0/W1C	0h	ADC Interrupt 1 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT1 and ADCINT1RESULT flags in the ADCINTFLG, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn

12.14.3.9 ADCINTOVF Register (Offset = 14h) [Reset = 00000000h]

ADCINTOVF is shown in Figure 12-47 and described in Table 12-48.

Return to the Summary Table.

ADC Interrupt Overflow Register

Figure 12-47. ADCINTOVF Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	ADCDMAINT2OVF	ADCDMAINT1OVF
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	ADCINT2OVF	ADCINT1OVF
R-0h				R-0h	R-0h	R-0h	R-0h

Table 12-48. ADCINTOVF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	ADCDMAINT2OVF	R	0h	ADC DMA Interrupt 2 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
16	ADCDMAINT1OVF	R	0h	ADC DMA Interrupt 1 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved

Table 12-48. ADCINTOVF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	ADCINT2OVF	R	0h	ADC Interrupt 2 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
0	ADCINT1OVF	R	0h	ADC Interrupt 1 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn

12.14.3.10 ADCINTOVFCLR Register (Offset = 16h) [Reset = 00000000h]

ADCINTOVFCLR is shown in Figure 12-48 and described in Table 12-49.

Return to the Summary Table.

ADC Interrupt Overflow Clear Register

Figure 12-48. ADCINTOVFCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	ADCDMAINT2OVF	ADCDMAINT1OVF
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	ADCINT2OVF	ADCINT1OVF
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 12-49. ADCINTOVFCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	RESERVED	R-0/W1C	0h	Reserved
18	RESERVED	R-0/W1C	0h	Reserved
17	ADCDMAINT2OVF	R-0/W1C	0h	ADC DMA Interrupt 2 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
16	ADCDMAINT1OVF	R-0/W1C	0h	ADC DMA Interrupt 1 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	RESERVED	R-0/W1C	0h	Reserved
2	RESERVED	R-0/W1C	0h	Reserved
1	ADCINT2OVF	R-0/W1C	0h	ADC Interrupt 2 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn

Table 12-49. ADCINTOVFCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ADCINT1OVF	R-0/W1C	0h	ADC Interrupt 1 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn

12.14.3.11 ADCSOCPRICTL Register (Offset = 18h) [Reset = 00000400h]

ADCSOCPRICTL is shown in Figure 12-49 and described in Table 12-50.

Return to the Summary Table.

ADC SOC Priority Control Register

Figure 12-49. ADCSOCPRICTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					RRPOINTER		
R-0h					R-10h		
7	6	5	4	3	2	1	0
RRPOINTER		RESERVED				SOCPRIORITY	
R-10h		R-0h				R/W-0h	

Table 12-50. ADCSOCPRICTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved

Table 12-50. ADCSOCPRICTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-6	RRPOINTER	R	10h	Round Robin Pointer. Holds the value of the last converted round robin SOCx to be used by the round robin scheme to determine order of conversions. 00h SOC0 was last round robin SOC to convert, SOC1 is highest round robin priority. 01h SOC1 was last round robin SOC to convert, SOC2 is highest round robin priority. 02h SOC2 was last round robin SOC to convert, SOC3 is highest round robin priority. 03h SOC3 was last round robin SOC to convert, SOC4 is highest round robin priority. 04h SOC4 was last round robin SOC to convert, SOC5 is highest round robin priority. 05h SOC5 was last round robin SOC to convert, SOC6 is highest round robin priority. 06h SOC6 was last round robin SOC to convert, SOC7 is highest round robin priority. 07h SOC7 was last round robin SOC to convert, SOC8 is highest round robin priority. 08h SOC8 was last round robin SOC to convert, SOC9 is highest round robin priority. 09h SOC9 was last round robin SOC to convert, SOC10 is highest round robin priority. 0Ah SOC10 was last round robin SOC to convert, SOC11 is highest round robin priority. 0Bh SOC11 was last round robin SOC to convert, SOC12 is highest round robin priority. 0Ch SOC12 was last round robin SOC to convert, SOC13 is highest round robin priority. 0Dh SOC13 was last round robin SOC to convert, SOC14 is highest round robin priority. 0Eh SOC14 was last round robin SOC to convert, SOC15 is highest round robin priority. 0Fh SOC15 was last round robin SOC to convert, SOC0 is highest round robin priority. 10h Reset value to indicate no SOC has been converted. SOC0 is highest round robin priority. Set to this value when the ADC module is reset by SOFTPRES or when the ADCSOCPRICTL register is written. In the latter case, if a conversion is currently in progress, it will complete and then the new priority will take effect. Others Invalid value. Reset type: SYSRSn
5-2	RESERVED	R	0h	Reserved
1-0	SOCPRIORITY	R/W	0h	SOC Priority Determines the cutoff point for priority mode and round robin arbitration for SOCx 0h SOC priority is handled in round robin mode for all channels. 1h SOC0 is high priority, rest of channels are in round robin mode. 2h SOC0-SOC1 are high priority, SOC2-SOC15 are in round robin mode. 3h SOC0-SOC2 are high priority, SOC3-SOC15 are in round robin mode. Reset type: SYSRSn

12.14.3.12 ADCINTSOCSEL1 Register (Offset = 1Ah) [Reset = 00000000h]

ADCINTSOCSEL1 is shown in Figure 12-50 and described in Table 12-51.

Return to the Summary Table.

ADC Interrupt SOC Selection 1 Register

Figure 12-50. ADCINTSOCSEL1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOC15		SOC14		SOC13		SOC12		SOC11		SOC10		SOC9		SOC8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC7		SOC6		SOC5		SOC4		SOC3		SOC2		SOC1		SOC0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 12-51. ADCINTSOCSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SOC15	R/W	0h	SOC15 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC15. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC15. TRIGSEL field alone determines SOC15 trigger. 01 ADCINT1 will trigger SOC15. 10 ADCINT2 will trigger SOC15. 11 EOC14 will trigger SOC15. Reset type: SYSRSn
29-28	SOC14	R/W	0h	SOC14 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC14. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC14. TRIGSEL field alone determines SOC14 trigger. 01 ADCINT1 will trigger SOC14. 10 ADCINT2 will trigger SOC14. 11 EOC13 will trigger SOC14. Reset type: SYSRSn
27-26	SOC13	R/W	0h	SOC13 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC13. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC13. TRIGSEL field alone determines SOC13 trigger. 01 ADCINT1 will trigger SOC13. 10 ADCINT2 will trigger SOC13. 11 EOC12 will trigger SOC13. Reset type: SYSRSn
25-24	SOC12	R/W	0h	SOC12 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC12. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC12. TRIGSEL field alone determines SOC12 trigger. 01 ADCINT1 will trigger SOC12. 10 ADCINT2 will trigger SOC12. 11 EOC11 will trigger SOC12. Reset type: SYSRSn

Table 12-51. ADCINTSOCSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23-22	SOC11	R/W	0h	SOC11 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC11. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC11. TRIGSEL field alone determines SOC11 trigger. 01 ADCINT1 will trigger SOC11. 10 ADCINT2 will trigger SOC11. 11 EOC10 will trigger SOC11. Reset type: SYSRSn
21-20	SOC10	R/W	0h	SOC10 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC10. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC10. TRIGSEL field alone determines SOC10 trigger. 01 ADCINT1 will trigger SOC10. 10 ADCINT2 will trigger SOC10. 11 EOC9 will trigger SOC10. Reset type: SYSRSn
19-18	SOC9	R/W	0h	SOC9 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC9. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC9. TRIGSEL field alone determines SOC9 trigger. 01 ADCINT1 will trigger SOC9. 10 ADCINT2 will trigger SOC1. 11 EOC8 will trigger SOC9. Reset type: SYSRSn
17-16	SOC8	R/W	0h	SOC8 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC8. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC8. TRIGSEL field alone determines SOC8 trigger. 01 ADCINT1 will trigger SOC8. 10 ADCINT2 will trigger SOC8. 11 EOC7 will trigger SOC8. Reset type: SYSRSn
15-14	SOC7	R/W	0h	SOC7 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC7. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC7. TRIGSEL field alone determines SOC7 trigger. 01 ADCINT1 will trigger SOC7. 10 ADCINT2 will trigger SOC7. 11 EOC6 will trigger SOC7. Reset type: SYSRSn
13-12	SOC6	R/W	0h	SOC6 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC6. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC6. TRIGSEL field alone determines SOC6 trigger. 01 ADCINT1 will trigger SOC6. 10 ADCINT2 will trigger SOC6. 11 EOC5 will trigger SOC6. Reset type: SYSRSn

Table 12-51. ADCINTSOCSEL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-10	SOC5	R/W	0h	SOC5 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC5. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC5. TRIGSEL field alone determines SOC5 trigger. 01 ADCINT1 will trigger SOC5. 10 ADCINT2 will trigger SOC5. 11 EOC4 will trigger SOC5. Reset type: SYSRSn
9-8	SOC4	R/W	0h	SOC4 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC4. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC4. TRIGSEL field alone determines SOC4 trigger. 01 ADCINT1 will trigger SOC4. 10 ADCINT2 will trigger SOC4. 11 EOC3 will trigger SOC4. Reset type: SYSRSn
7-6	SOC3	R/W	0h	SOC3 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC3. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC3. TRIGSEL field alone determines SOC3 trigger. 01 ADCINT1 will trigger SOC3. 10 ADCINT2 will trigger SOC3. 11 EOC2 will trigger SOC3. Reset type: SYSRSn
5-4	SOC2	R/W	0h	SOC2 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC2. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC2. TRIGSEL field alone determines SOC2 trigger. 01 ADCINT1 will trigger SOC2. 10 ADCINT2 will trigger SOC2. 11 EOC1 will trigger SOC2. Reset type: SYSRSn
3-2	SOC1	R/W	0h	SOC1 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC1. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC1. TRIGSEL field alone determines SOC1 trigger. 01 ADCINT1 will trigger SOC1. 10 ADCINT2 will trigger SOC1. 11 EOC0 will trigger SOC1. Reset type: SYSRSn
1-0	SOC0	R/W	0h	SOC0 ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOC0. The trigger selected in this field is in addition to the TRIGSEL field in the ADCSOCxCTL register. 00 No ADCINT will trigger SOC0. TRIGSEL field alone determines SOC0 trigger. 01 ADCINT1 will trigger SOC0. 10 ADCINT2 will trigger SOC0. 11 EOC[N-1] will trigger SOC0. N - Total number of SOCs supported Reset type: SYSRSn

12.14.3.13 ADCSOCFLG1 Register (Offset = 1Eh) [Reset = 00000000h]

ADCSOCFLG1 is shown in Figure 12-51 and described in Table 12-52.

Return to the Summary Table.

ADC SOC Flag 1 Register

Figure 12-51. ADCSOCFLG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 12-52. ADCSOCFLG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15	R	0h	SOC15 Start of Conversion Flag. Indicates the state of SOC15 conversions. 0 No sample pending for SOC15. 1 Trigger has been received and sample is pending for SOC15. This bit will be automatically cleared when the SOC15 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
14	SOC14	R	0h	SOC14 Start of Conversion Flag. Indicates the state of SOC14 conversions. 0 No sample pending for SOC14. 1 Trigger has been received and sample is pending for SOC14. This bit will be automatically cleared when the SOC14 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 12-52. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	SOC13	R	0h	SOC13 Start of Conversion Flag. Indicates the state of SOC13 conversions. 0 No sample pending for SOC13. 1 Trigger has been received and sample is pending for SOC13. This bit will be automatically cleared when the SOC13 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
12	SOC12	R	0h	SOC12 Start of Conversion Flag. Indicates the state of SOC12 conversions. 0 No sample pending for SOC12. 1 Trigger has been received and sample is pending for SOC12. This bit will be automatically cleared when the SOC12 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
11	SOC11	R	0h	SOC11 Start of Conversion Flag. Indicates the state of SOC11 conversions. 0 No sample pending for SOC11. 1 Trigger has been received and sample is pending for SOC11. This bit will be automatically cleared when the SOC11 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
10	SOC10	R	0h	SOC10 Start of Conversion Flag. Indicates the state of SOC10 conversions. 0 No sample pending for SOC10. 1 Trigger has been received and sample is pending for SOC10. This bit will be automatically cleared when the SOC10 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
9	SOC9	R	0h	SOC9 Start of Conversion Flag. Indicates the state of SOC9 conversions. 0 No sample pending for SOC9. 1 Trigger has been received and sample is pending for SOC9. This bit will be automatically cleared when the SOC9 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 12-52. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SOC8	R	0h	SOC8 Start of Conversion Flag. Indicates the state of SOC8 conversions. 0 No sample pending for SOC8. 1 Trigger has been received and sample is pending for SOC8. This bit will be automatically cleared when the SOC8 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
7	SOC7	R	0h	SOC7 Start of Conversion Flag. Indicates the state of SOC7 conversions. 0 No sample pending for SOC7. 1 Trigger has been received and sample is pending for SOC7. This bit will be automatically cleared when the SOC7 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
6	SOC6	R	0h	SOC6 Start of Conversion Flag. Indicates the state of SOC6 conversions. 0 No sample pending for SOC6. 1 Trigger has been received and sample is pending for SOC6. This bit will be automatically cleared when the SOC6 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
5	SOC5	R	0h	SOC5 Start of Conversion Flag. Indicates the state of SOC5 conversions. 0 No sample pending for SOC5. 1 Trigger has been received and sample is pending for SOC5. This bit will be automatically cleared when the SOC5 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
4	SOC4	R	0h	SOC4 Start of Conversion Flag. Indicates the state of SOC4 conversions. 0 No sample pending for SOC4. 1 Trigger has been received and sample is pending for SOC4. This bit will be automatically cleared when the SOC4 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 12-52. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	SOC3	R	0h	SOC3 Start of Conversion Flag. Indicates the state of SOC3 conversions. 0 No sample pending for SOC3. 1 Trigger has been received and sample is pending for SOC3. This bit will be automatically cleared when the SOC3 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
2	SOC2	R	0h	SOC2 Start of Conversion Flag. Indicates the state of SOC2 conversions. 0 No sample pending for SOC2. 1 Trigger has been received and sample is pending for SOC2. This bit will be automatically cleared when the SOC2 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
1	SOC1	R	0h	SOC1 Start of Conversion Flag. Indicates the state of SOC1 conversions. 0 No sample pending for SOC1. 1 Trigger has been received and sample is pending for SOC1. This bit will be automatically cleared when the SOC1 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
0	SOC0	R	0h	SOC0 Start of Conversion Flag. Indicates the state of SOC0 conversions. 0 No sample pending for SOC0. 1 Trigger has been received and sample is pending for SOC0. This bit will be automatically cleared when the SOC0 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

12.14.3.14 ADCSOCFRC1 Register (Offset = 20h) [Reset = 00000000h]

ADCSOCFRC1 is shown in Figure 12-52 and described in Table 12-53.

Return to the Summary Table.

ADC SOC Force 1 Register

Figure 12-52. ADCSOCFRC1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 12-53. ADCSOCFRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15	R-0/W1S	0h	SOC15 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC15 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC15 flag bit to 1. This will cause a conversion to start once priority is given to SOC15. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC15 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
14	SOC14	R-0/W1S	0h	SOC14 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC14 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC14 flag bit to 1. This will cause a conversion to start once priority is given to SOC14. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC14 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 12-53. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	SOC13	R-0/W1S	0h	SOC13 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC13 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC13 flag bit to 1. This will cause a conversion to start once priority is given to SOC13. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC13 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
12	SOC12	R-0/W1S	0h	SOC12 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC12 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC12 flag bit to 1. This will cause a conversion to start once priority is given to SOC12. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC12 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
11	SOC11	R-0/W1S	0h	SOC11 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC11 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC11 flag bit to 1. This will cause a conversion to start once priority is given to SOC11. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC11 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
10	SOC10	R-0/W1S	0h	SOC10 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC10 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC10 flag bit to 1. This will cause a conversion to start once priority is given to SOC10. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC10 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 12-53. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	SOC9	R-0/W1S	0h	SOC9 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC9 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC9 flag bit to 1. This will cause a conversion to start once priority is given to SOC9. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC9 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
8	SOC8	R-0/W1S	0h	SOC8 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC8 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC8 flag bit to 1. This will cause a conversion to start once priority is given to SOC8. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC8 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
7	SOC7	R-0/W1S	0h	SOC7 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC7 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC7 flag bit to 1. This will cause a conversion to start once priority is given to SOC7. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC7 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
6	SOC6	R-0/W1S	0h	SOC6 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC6 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC6 flag bit to 1. This will cause a conversion to start once priority is given to SOC6. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC6 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 12-53. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	SOC5	R-0/W1S	0h	SOC5 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC5 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC5 flag bit to 1. This will cause a conversion to start once priority is given to SOC5. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC5 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
4	SOC4	R-0/W1S	0h	SOC4 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC4 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC4 flag bit to 1. This will cause a conversion to start once priority is given to SOC4. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC4 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
3	SOC3	R-0/W1S	0h	SOC3 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC3 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC3 flag bit to 1. This will cause a conversion to start once priority is given to SOC3. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC3 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
2	SOC2	R-0/W1S	0h	SOC2 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC2 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC2 flag bit to 1. This will cause a conversion to start once priority is given to SOC2. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC2 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

Table 12-53. ADCSOCFRC1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SOC1	R-0/W1S	0h	SOC1 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC1 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC1 flag bit to 1. This will cause a conversion to start once priority is given to SOC1. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC1 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn
0	SOC0	R-0/W1S	0h	SOC0 Force Start of Conversion Bit. Writing a 1 will force to 1 the SOC0 flag in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored. This bit will always read as a 0. 0 No action. 1 Force SOC0 flag bit to 1. This will cause a conversion to start once priority is given to SOC0. If software tries to set this bit on the same clock cycle that hardware tries to clear the SOC0 bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not. Reset type: SYSRSn

12.14.3.15 ADCSOCOVF1 Register (Offset = 22h) [Reset = 00000000h]

ADCSOCOVF1 is shown in Figure 12-53 and described in Table 12-54.

Return to the Summary Table.

ADC SOC Overflow 1 Register

Figure 12-53. ADCSOCOVF1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
SOC15OVF	SOC14OVF	SOC13OVF	SOC12OVF	SOC11OVF	SOC10OVF	SOC9OVF	SOC8OVF
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SOC7OVF	SOC6OVF	SOC5OVF	SOC4OVF	SOC3OVF	SOC2OVF	SOC1OVF	SOC0OVF
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 12-54. ADCSOCOVF1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15OVF	R	0h	SOC15 Start of Conversion Overflow Flag. Indicates an SOC15 event was generated in hardware while an existing SOC15 event was already pending. 0 No SOC15 event overflow. 1 SOC15 event overflow. An overflow condition does not stop SOC15 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
14	SOC14OVF	R	0h	SOC14 Start of Conversion Overflow Flag. Indicates an SOC14 event was generated in hardware while an existing SOC14 event was already pending. 0 No SOC14 event overflow. 1 SOC14 event overflow. An overflow condition does not stop SOC14 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
13	SOC13OVF	R	0h	SOC13 Start of Conversion Overflow Flag. Indicates an SOC13 event was generated in hardware while an existing SOC13 event was already pending. 0 No SOC13 event overflow. 1 SOC13 event overflow. An overflow condition does not stop SOC13 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

Table 12-54. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	SOC12OVF	R	0h	SOC12 Start of Conversion Overflow Flag. Indicates an SOC12 event was generated in hardware while an existing SOC12 event was already pending. 0 No SOC12 event overflow. 1 SOC12 event overflow. An overflow condition does not stop SOC12 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
11	SOC11OVF	R	0h	SOC11 Start of Conversion Overflow Flag. Indicates an SOC11 event was generated in hardware while an existing SOC11 event was already pending. 0 No SOC11 event overflow. 1 SOC11 event overflow. An overflow condition does not stop SOC11 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
10	SOC10OVF	R	0h	SOC10 Start of Conversion Overflow Flag. Indicates an SOC10 event was generated in hardware while an existing SOC10 event was already pending. 0 No SOC10 event overflow. 1 SOC10 event overflow. An overflow condition does not stop SOC10 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
9	SOC9OVF	R	0h	SOC9 Start of Conversion Overflow Flag. Indicates an SOC9 event was generated in hardware while an existing SOC9 event was already pending. 0 No SOC9 event overflow. 1 SOC9 event overflow. An overflow condition does not stop SOC9 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
8	SOC8OVF	R	0h	SOC8 Start of Conversion Overflow Flag. Indicates an SOC8 event was generated in hardware while an existing SOC8 event was already pending. 0 No SOC8 event overflow. 1 SOC8 event overflow. An overflow condition does not stop SOC8 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
7	SOC7OVF	R	0h	SOC7 Start of Conversion Overflow Flag. Indicates an SOC7 event was generated in hardware while an existing SOC7 event was already pending. 0 No SOC7 event overflow. 1 SOC7 event overflow. An overflow condition does not stop SOC7 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

Table 12-54. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	SOC6OVF	R	0h	SOC6 Start of Conversion Overflow Flag. Indicates an SOC6 event was generated in hardware while an existing SOC6 event was already pending. 0 No SOC6 event overflow. 1 SOC6 event overflow. An overflow condition does not stop SOC6 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
5	SOC5OVF	R	0h	SOC5 Start of Conversion Overflow Flag. Indicates an SOC5 event was generated in hardware while an existing SOC5 event was already pending. 0 No SOC5 event overflow. 1 SOC5 event overflow. An overflow condition does not stop SOC5 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
4	SOC4OVF	R	0h	SOC4 Start of Conversion Overflow Flag. Indicates an SOC4 event was generated in hardware while an existing SOC4 event was already pending. 0 No SOC4 event overflow. 1 SOC4 event overflow. An overflow condition does not stop SOC4 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
3	SOC3OVF	R	0h	SOC3 Start of Conversion Overflow Flag. Indicates an SOC3 event was generated in hardware while an existing SOC3 event was already pending. 0 No SOC3 event overflow. 1 SOC3 event overflow. An overflow condition does not stop SOC3 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
2	SOC2OVF	R	0h	SOC2 Start of Conversion Overflow Flag. Indicates an SOC2 event was generated in hardware while an existing SOC2 event was already pending. 0 No SOC2 event overflow. 1 SOC2 event overflow. An overflow condition does not stop SOC2 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn
1	SOC1OVF	R	0h	SOC1 Start of Conversion Overflow Flag. Indicates an SOC1 event was generated in hardware while an existing SOC1 event was already pending. 0 No SOC1 event overflow. 1 SOC1 event overflow. An overflow condition does not stop SOC1 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

Table 12-54. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SOC0OVF	R	0h	SOC0 Start of Conversion Overflow Flag. Indicates an SOC0 event was generated in hardware while an existing SOC0 event was already pending. 0 No SOC0 event overflow. 1 SOC0 event overflow. An overflow condition does not stop SOC0 events from being processed. It simply is an indication that a hardware trigger was missed. A write to the ADCSOCFRC1 register does not affect this bit. Reset type: SYSRSn

12.14.3.16 ADCSOCOVFCLR1 Register (Offset = 24h) [Reset = 00000000h]

ADCSOCOVFCLR1 is shown in Figure 12-54 and described in Table 12-55.

Return to the Summary Table.

ADC SOC Overflow Clear 1 Register

Figure 12-54. ADCSOCOVFCLR1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
SOC15OVF	SOC14OVF	SOC13OVF	SOC12OVF	SOC11OVF	SOC10OVF	SOC9OVF	SOC8OVF
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
SOC7OVF	SOC6OVF	SOC5OVF	SOC4OVF	SOC3OVF	SOC2OVF	SOC1OVF	SOC0OVF
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 12-55. ADCSOCOVFCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15OVF	R-0/W1C	0h	SOC15 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC15 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC15 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
14	SOC14OVF	R-0/W1C	0h	SOC14 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC14 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC14 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
13	SOC13OVF	R-0/W1C	0h	SOC13 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC13 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC13 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 12-55. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	SOC12OVF	R-0/W1C	0h	SOC12 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC12 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC12 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
11	SOC11OVF	R-0/W1C	0h	SOC11 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC11 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC11 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
10	SOC10OVF	R-0/W1C	0h	SOC10 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC10 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC10 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
9	SOC9OVF	R-0/W1C	0h	SOC9 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC9 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC9 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
8	SOC8OVF	R-0/W1C	0h	SOC8 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC8 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC8 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
7	SOC7OVF	R-0/W1C	0h	SOC7 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC7 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC7 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 12-55. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	SOC6OVF	R-0/W1C	0h	SOC6 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC6 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC6 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
5	SOC5OVF	R-0/W1C	0h	SOC5 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC5 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC5 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
4	SOC4OVF	R-0/W1C	0h	SOC4 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC4 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC4 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
3	SOC3OVF	R-0/W1C	0h	SOC3 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC3 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC3 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
2	SOC2OVF	R-0/W1C	0h	SOC2 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC2 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC2 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
1	SOC1OVF	R-0/W1C	0h	SOC1 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC1 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC1 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 12-55. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SOC0OVF	R-0/W1C	0h	SOC0 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC0 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC0 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

12.14.3.17 ADCSOC0CTL Register (Offset = 26h) [Reset = 00000200h]

ADCSOC0CTL is shown in Figure 12-55 and described in Table 12-56.

Return to the Summary Table.

ADC SOC0 Control Register

Figure 12-55. ADCSOC0CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-56. ADCSOC0CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC0 Trigger Source Select. Along with the SOC0 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC0 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC0 Channel Select. Selects the channel to be converted when SOC0 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC0 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC0 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-56. ADCSOC0CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC0 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.18 ADCSOC1CTL Register (Offset = 28h) [Reset = 00000200h]

ADCSOC1CTL is shown in Figure 12-56 and described in Table 12-57.

Return to the Summary Table.

ADC SOC1 Control Register

Figure 12-56. ADCSOC1CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-57. ADCSOC1CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC1 Trigger Source Select. Along with the SOC1 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC1 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC1 Channel Select. Selects the channel to be converted when SOC1 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC1 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC1 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-57. ADCSOC1CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC1 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.19 ADCSOC2CTL Register (Offset = 2Ah) [Reset = 00000200h]

ADCSOC2CTL is shown in Figure 12-57 and described in Table 12-58.

Return to the Summary Table.

ADC SOC2 Control Register

Figure 12-57. ADCSOC2CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-58. ADCSOC2CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC2 Trigger Source Select. Along with the SOC2 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC2 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC2 Channel Select. Selects the channel to be converted when SOC2 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC2 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC2 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-58. ADCSOC2CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC2 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.20 ADCSOC3CTL Register (Offset = 2Ch) [Reset = 00000200h]

ADCSOC3CTL is shown in Figure 12-58 and described in Table 12-59.

Return to the Summary Table.

ADC SOC3 Control Register

Figure 12-58. ADCSOC3CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-59. ADCSOC3CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC3 Trigger Source Select. Along with the SOC3 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC3 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC3 Channel Select. Selects the channel to be converted when SOC3 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC3 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC3 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-59. ADCSOC3CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC3 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.21 ADCSOC4CTL Register (Offset = 2Eh) [Reset = 00000200h]

ADCSOC4CTL is shown in Figure 12-59 and described in Table 12-60.

Return to the Summary Table.

ADC SOC4 Control Register

Figure 12-59. ADCSOC4CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-60. ADCSOC4CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC4 Trigger Source Select. Along with the SOC4 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC4 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC4 Channel Select. Selects the channel to be converted when SOC4 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC4 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC4 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-60. ADCSOC4CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC4 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.22 ADCSOC5CTL Register (Offset = 30h) [Reset = 00000200h]

ADCSOC5CTL is shown in Figure 12-60 and described in Table 12-61.

Return to the Summary Table.

ADC SOC5 Control Register

Figure 12-60. ADCSOC5CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-61. ADCSOC5CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC5 Trigger Source Select. Along with the SOC5 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC5 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC5 Channel Select. Selects the channel to be converted when SOC5 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC5 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC5 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-61. ADCSOC5CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC5 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.23 ADCSOC6CTL Register (Offset = 32h) [Reset = 00000200h]

ADCSOC6CTL is shown in Figure 12-61 and described in Table 12-62.

Return to the Summary Table.

ADC SOC6 Control Register

Figure 12-61. ADCSOC6CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-62. ADCSOC6CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC6 Trigger Source Select. Along with the SOC6 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC6 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC6 Channel Select. Selects the channel to be converted when SOC6 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC6 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC6 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-62. ADCSOC6CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC6 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.24 ADCSOC7CTL Register (Offset = 34h) [Reset = 00000200h]

ADCSOC7CTL is shown in Figure 12-62 and described in Table 12-63.

Return to the Summary Table.

ADC SOC7 Control Register

Figure 12-62. ADCSOC7CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-63. ADCSOC7CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC7 Trigger Source Select. Along with the SOC7 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC7 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC7 Channel Select. Selects the channel to be converted when SOC7 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC7 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC7 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-63. ADCSOC7CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC7 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.25 ADCSOC8CTL Register (Offset = 36h) [Reset = 00000200h]

ADCSOC8CTL is shown in Figure 12-63 and described in Table 12-64.

Return to the Summary Table.

ADC SOC8 Control Register

Figure 12-63. ADCSOC8CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-64. ADCSOC8CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC8 Trigger Source Select. Along with the SOC8 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC8 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC8 Channel Select. Selects the channel to be converted when SOC8 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC8 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC8 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-64. ADCSOC8CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC8 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.26 ADCSOC9CTL Register (Offset = 38h) [Reset = 00000200h]

ADCSOC9CTL is shown in Figure 12-64 and described in Table 12-65.

Return to the Summary Table.

ADC SOC9 Control Register

Figure 12-64. ADCSOC9CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-65. ADCSOC9CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC9 Trigger Source Select. Along with the SOC9 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC9 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC9 Channel Select. Selects the channel to be converted when SOC9 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC9 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC9 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-65. ADCSOC9CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC9 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.27 ADCSOC10CTL Register (Offset = 3Ah) [Reset = 00000200h]

ADCSOC10CTL is shown in Figure 12-65 and described in Table 12-66.

Return to the Summary Table.

ADC SOC10 Control Register

Figure 12-65. ADCSOC10CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-66. ADCSOC10CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC10 Trigger Source Select. Along with the SOC10 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC10 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC10 Channel Select. Selects the channel to be converted when SOC10 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC10 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC10 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-66. ADCSOC10CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC10 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.28 ADCSOC11CTL Register (Offset = 3Ch) [Reset = 00000200h]

ADCSOC11CTL is shown in Figure 12-66 and described in Table 12-67.

Return to the Summary Table.

ADC SOC11 Control Register

Figure 12-66. ADCSOC11CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-67. ADCSOC11CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC11 Trigger Source Select. Along with the SOC11 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC11 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC11 Channel Select. Selects the channel to be converted when SOC11 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC11 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC11 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-67. ADCSOC11CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC11 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.29 ADCSOC12CTL Register (Offset = 3Eh) [Reset = 00000200h]

ADCSOC12CTL is shown in Figure 12-67 and described in Table 12-68.

Return to the Summary Table.

ADC SOC12 Control Register

Figure 12-67. ADCSOC12CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-68. ADCSOC12CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC12 Trigger Source Select. Along with the SOC12 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC12 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC12 Channel Select. Selects the channel to be converted when SOC12 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC12 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC12 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-68. ADCSOC12CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC12 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.30 ADCSOC13CTL Register (Offset = 40h) [Reset = 00000200h]

ADCSOC13CTL is shown in Figure 12-68 and described in Table 12-69.

Return to the Summary Table.

ADC SOC13 Control Register

Figure 12-68. ADCSOC13CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-69. ADCSOC13CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC13 Trigger Source Select. Along with the SOC13 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC13 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC13 Channel Select. Selects the channel to be converted when SOC13 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC13 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC13 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-69. ADCSOC13CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC13 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.31 ADCSOC14CTL Register (Offset = 42h) [Reset = 00000200h]

ADCSOC14CTL is shown in Figure 12-69 and described in Table 12-70.

Return to the Summary Table.

ADC SOC14 Control Register

Figure 12-69. ADCSOC14CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-70. ADCSOC14CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC14 Trigger Source Select. Along with the SOC14 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC14 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC14 Channel Select. Selects the channel to be converted when SOC14 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC14 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC14 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-70. ADCSOC14CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC14 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.32 ADCSOC15CTL Register (Offset = 44h) [Reset = 00000200h]

ADCSOC15CTL is shown in Figure 12-70 and described in Table 12-71.

Return to the Summary Table.

ADC SOC15 Control Register

Figure 12-70. ADCSOC15CTL Register

31	30	29	28	27	26	25	24
RESERVED							TRIGSEL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				CHSEL			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
CHSEL	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 12-71. ADCSOC15CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC15 Trigger Source Select. Along with the SOC15 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC15 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 00h ADCTRIG0 - Software only 01h - 1Fh Hardware triggers Reset type: SYSRSn
19-15	CHSEL	R/W	0h	SOC15 Channel Select. Selects the channel to be converted when SOC15 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC15 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC15 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn

Table 12-71. ADCSOC15CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC15 Acquisition Prescale. Controls the sample and hold window for this SOC. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is '00' S+H window = ACQPS[5:0] + 1 sysclk cycles '01' S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles '10' S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles '11' S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

12.14.3.33 ADCEVTSTAT Register (Offset = 66h) [Reset = 00000000h]

ADCEVTSTAT is shown in Figure 12-71 and described in Table 12-72.

Return to the Summary Table.

ADC Event Status Register

Figure 12-71. ADCEVTSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	PPB3INLIMIT	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
PPB2INLIMIT	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	PPB1INLIMIT	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 12-72. ADCEVTSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	PPB3INLIMIT	R	0h	Post Processing Block 3 In-Limit Trip Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
10	PPB3ZERO	R	0h	Post Processing Block 3 Zero Crossing Flag. When set indicates the ADCPPB3RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 12-72. ADCEVTSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	PPB3TRIPLO	R	0h	Post Processing Block 3 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
8	PPB3TRIPHI	R	0h	Post Processing Block 3 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
7	PPB2INLIMIT	R	0h	Post Processing Block 2 In-Limit Trip Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
6	PPB2ZERO	R	0h	Post Processing Block 2 Zero Crossing Flag. When set indicates the ADCPPB2RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
5	PPB2TRIPLO	R	0h	Post Processing Block 2 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
4	PPB2TRIPHI	R	0h	Post Processing Block 2 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 12-72. ADCEVTSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PPB1INLIMIT	R	0h	Post Processing Block 1 In-Limit Trip Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
2	PPB1ZERO	R	0h	Post Processing Block 1 Zero Crossing Flag. When set indicates the ADCPPB1RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
1	PPB1TRIPLO	R	0h	Post Processing Block 1 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
0	PPB1TRIPHI	R	0h	Post Processing Block 1 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

12.14.3.34 ADCEVTCLR Register (Offset = 68h) [Reset = 00000000h]

ADCEVTCLR is shown in Figure 12-72 and described in Table 12-73.

Return to the Summary Table.

ADC Event Clear Register

Figure 12-72. ADCEVTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	PPB3INLIMIT	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
PPB2INLIMIT	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	PPB1INLIMIT	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 12-73. ADCEVTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R-0/W1C	0h	Reserved
14	RESERVED	R-0/W1C	0h	Reserved
13	RESERVED	R-0/W1C	0h	Reserved
12	RESERVED	R-0/W1C	0h	Reserved
11	PPB3INLIMIT	R-0/W1C	0h	Post Processing Block 3 In-Limit Trip Flag Clear. Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
10	PPB3ZERO	R-0/W1C	0h	Post Processing Block 3 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
9	PPB3TRIPLO	R-0/W1C	0h	Post Processing Block 3 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
8	PPB3TRIPHI	R-0/W1C	0h	Post Processing Block 3 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
7	PPB2INLIMIT	R-0/W1C	0h	Post Processing Block 2 In-Limit Trip Flag Clear. Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 12-73. ADCEVTCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	PPB2ZERO	R-0/W1C	0h	Post Processing Block 2 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
5	PPB2TRIPLO	R-0/W1C	0h	Post Processing Block 2 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
4	PPB2TRIPHI	R-0/W1C	0h	Post Processing Block 2 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
3	PPB1INLIMIT	R-0/W1C	0h	Post Processing Block 1 In-Limit Trip Flag Clear. Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
2	PPB1ZERO	R-0/W1C	0h	Post Processing Block 1 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
1	PPB1TRIPLO	R-0/W1C	0h	Post Processing Block 1 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
0	PPB1TRIPHI	R-0/W1C	0h	Post Processing Block 1 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

12.14.3.35 ADCEVTSEL Register (Offset = 6Ah) [Reset = 00000000h]

ADCEVTSEL is shown in Figure 12-73 and described in Table 12-74.

Return to the Summary Table.

ADC Event Selection Register

Figure 12-73. ADCEVTSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	PPB3INLIMIT	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PPB2INLIMIT	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	PPB1INLIMIT	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 12-74. ADCEVTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	PPB3INLIMIT	R/W	0h	Post Processing Block 3 In-Limit Trip Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
10	PPB3ZERO	R/W	0h	Post Processing Block 3 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
9	PPB3TRIPLO	R/W	0h	Post Processing Block 3 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
8	PPB3TRIPHI	R/W	0h	Post Processing Block 3 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
7	PPB2INLIMIT	R/W	0h	Post Processing Block 2 In-Limit Trip Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn

Table 12-74. ADCEVTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	PPB2ZERO	R/W	0h	Post Processing Block 2 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
5	PPB2TRIPLO	R/W	0h	Post Processing Block 2 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
4	PPB2TRIPHI	R/W	0h	Post Processing Block 2 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
3	PPB1INLIMIT	R/W	0h	Post Processing Block 1 In-Limit Trip Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
2	PPB1ZERO	R/W	0h	Post Processing Block 1 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
1	PPB1TRIPLO	R/W	0h	Post Processing Block 1 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
0	PPB1TRIPHI	R/W	0h	Post Processing Block 1 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn

12.14.3.36 ADCEVTINTSEL Register (Offset = 6Ch) [Reset = 00000000h]

ADCEVTINTSEL is shown in Figure 12-74 and described in Table 12-75.

Return to the Summary Table.

ADC Event Interrupt Selection Register

Figure 12-74. ADCEVTINTSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	PPB3INLIMIT	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PPB2INLIMIT	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	PPB1INLIMIT	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 12-75. ADCEVTINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	PPB3INLIMIT	R/W	0h	Post Processing Block 3 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
10	PPB3ZERO	R/W	0h	Post Processing Block 3 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
9	PPB3TRIPLO	R/W	0h	Post Processing Block 3 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
8	PPB3TRIPHI	R/W	0h	Post Processing Block 3 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
7	PPB2INLIMIT	R/W	0h	Post Processing Block 2 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn

Table 12-75. ADCEVTINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	PPB2ZERO	R/W	0h	Post Processing Block 2 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
5	PPB2TRIPLO	R/W	0h	Post Processing Block 2 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
4	PPB2TRIPHI	R/W	0h	Post Processing Block 2 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
3	PPB1INLIMIT	R/W	0h	Post Processing Block 1 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
2	PPB1ZERO	R/W	0h	Post Processing Block 1 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
1	PPB1TRIPLO	R/W	0h	Post Processing Block 1 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
0	PPB1TRIPHI	R/W	0h	Post Processing Block 1 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn

12.14.3.37 ADCREV Register (Offset = 72h) [Reset = 00000006h]

ADCREV is shown in Figure 12-75 and described in Table 12-76.

Return to the Summary Table.

ADC Revision Register

Figure 12-75. ADCREV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																								TYPE							
R-0h																								R-6h							

Table 12-76. ADCREV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	REV	R	0h	ADC Revision. To allow documentation of differences between revisions. First version is labeled as 00h. Reset type: SYSRSn
7-0	TYPE	R	6h	ADC Type. Always set to 6 for this HSADC-12b. Reset type: SYSRSn

12.14.3.38 ADCOFFTRIM Register (Offset = 74h) [Reset = 00000000h]

ADCOFFTRIM is shown in Figure 12-76 and described in Table 12-77.

Return to the Summary Table.

ADC Offset Trim Register 1

Figure 12-76. ADCOFFTRIM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OFFTRIM							
R-0h																								R/W-0h							

Table 12-77. ADCOFFTRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OFFTRIM	R/W	0h	ADC Offset Trim Adjusts the conversion results of the converter up or down to account for offset error in the ADC. A factory trim setting will be loaded during device boot. Offset can be corrected in the range of +7 to -8 LSBs. Value is 16*Offset in 8-bit 2's complement: 7 LSB (16*7) = 112 6 LSB (16*6) = 96 5 LSB (16*5) = 80 4 LSB (16*4) = 64 3 LSB (16*3) = 48 2 LSB (16*2) = 32 1 LSB (16*1) = 16 0 LSB (16*0) = 0 -1 LSB (16*(-1)) = 240 : : -7LSB(16*(-7)) = 144 Reset type: XRSn

12.14.3.39 ADCPPB1CONFIG Register (Offset = 80h) [Reset = 00000000h]

ADCPPB1CONFIG is shown in Figure 12-77 and described in Table 12-78.

Return to the Summary Table.

ADC PPB{#} Config Register

Figure 12-77. ADCPPB1CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TWOSCOMPEN	RESERVED	CBCEN	RESERVED	CONFIG			
R/W-0h	R-0h	R/W-0h	R-0h	R/W-0h			

Table 12-78. ADCPPB1CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TWOSCOMPEN	R/W	0h	ADC Post Processing Block {#} Two's Complement Enable. When set this bit enables the post conversion hardware processing circuit that performs a two's complement on the output of the offset/reference subtraction unit before storing the result in the ADCPPB{#}RESULT register. 0 ADCPPB{#}RESULT = ADCRESULTx - ADCPPB{#}OFFREF 1 ADCPPB{#}RESULT = ADCPPB{#}OFFREF - ADCRESULTx Reset type: SYSRSn
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4	RESERVED	R	0h	Reserved

Table 12-78. ADCPPB1CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	CONFIG	R/W	0h	ADC Post Processing Block {#} Configuration. This bit field defines which SOC/EOC/RESULT is associated with this post processing block. 0000 SOC0/EOC0/RESULT0 is associated with post processing block {#} 0001 SOC1/EOC1/RESULT1 is associated with post processing block {#} 0010 SOC2/EOC2/RESULT2 is associated with post processing block {#} 0011 SOC3/EOC3/RESULT3 is associated with post processing block {#} 0100 SOC4/EOC4/RESULT4 is associated with post processing block {#} 0101 SOC5/EOC5/RESULT5 is associated with post processing block {#} 0110 SOC6/EOC6/RESULT6 is associated with post processing block {#} 0111 SOC7/EOC7/RESULT7 is associated with post processing block {#} 1000 SOC8/EOC8/RESULT8 is associated with post processing block {#} 1001 SOC9/EOC9/RESULT9 is associated with post processing block {#} 1010 SOC10/EOC10/RESULT10 is associated with post processing block {#} 1011 SOC11/EOC11/RESULT11 is associated with post processing block {#} 1100 SOC12/EOC12/RESULT12 is associated with post processing block {#} 1101 SOC13/EOC13/RESULT13 is associated with post processing block {#} 1110 SOC14/EOC14/RESULT14 is associated with post processing block {#} 1111 SOC15/EOC15/RESULT15 is associated with post processing block {#} Reset type: SYSRSn

12.14.3.40 ADCPPB1OFFCAL Register (Offset = 84h) [Reset = 00000000h]

ADCPPB1OFFCAL is shown in Figure 12-78 and described in Table 12-79.

Return to the Summary Table.

ADC PPB1 Offset Calibration Register

Figure 12-78. ADCPPB1OFFCAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										OFFCAL					
R-0h																										R/W-0h					

Table 12-79. ADCPPB1OFFCAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	OFFCAL	R/W	0h	ADC Post Processing Block 1 Offset Correction. This bit field can be used to digitally remove any system level offset inherent in the ADCIN circuit. This 6-bit signed value is subtracted from the ADC output before being stored in the ADCRESULT register. 00h No change. The ADC output is stored directly into ADCRESULT. 01h ADC output - 1 is stored into ADCRESULT. 02h ADC output - 2 is stored into ADCRESULT. ... 20h ADC output + 32 is stored into ADCRESULT. ... 3Fh ADC output + 1 is stored into ADCRESULT. NOTE: The subtraction will saturate at 0000h and 0FFFh before being stored into the ADCRESULT register. Note: in the case that multiple PPBs point to the same SOC, only the OFFCAL of the lowest numbered PPB will be applied. Reset type: SYSRSn

12.14.3.41 ADCPPB1OFFREF Register (Offset = 86h) [Reset = 00000000h]

ADCPPB1OFFREF is shown in Figure 12-79 and described in Table 12-80.

Return to the Summary Table.

ADC PPB1 Offset Reference Register

Figure 12-79. ADCPPB1OFFREF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				OFFREF											
R-0h																				R/W-0h											

Table 12-80. ADCPPB1OFFREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-0	OFFREF	R/W	0h	ADC Post Processing Block 1 Offset Correction. This bit field can be used to either calculate the feedback error or convert a unipolar signal to bipolar by subtracting a reference value. This 12-bit unsigned value is subtracted from the ADCRESULT register before being passed through an optional two's complement function and stored in the ADCPPB1RESULT register. This subtraction is not saturated. 0000h No change. The ADCRESULT value is passed on. 001h ADCRESULT - 1 is passed on. 002h ADCRESULT - 2 is passed on. ... 800h ADCRESULT - 2048 is passed on. ... FFFh ADCRESULT - 4096 is passed on. Reset type: SYSRSn

12.14.3.42 ADCPPB1TRIPHI Register (Offset = 88h) [Reset = 00000000h]

ADCPPB1TRIPHI is shown in Figure 12-80 and described in Table 12-81.

Return to the Summary Table.

ADC PPB1 Trip High Register

Figure 12-80. ADCPPB1TRIPHI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LIMITHI															
R-0h																R/W-0h															

Table 12-81. ADCPPB1TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITHI	R/W	0h	ADC Post Processing Block 1 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[15:13] will be ignored Reset type: SYSRSn

12.14.3.43 ADCPPB1TRIPLO Register (Offset = 8Ah) [Reset = 00000000h]

ADCPPB1TRIPLO is shown in Figure 12-81 and described in Table 12-82.

Return to the Summary Table.

ADC PPB1 Trip Low/Trigger Time Stamp Register

Figure 12-81. ADCPPB1TRIPLO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LIMITLO															
R-0h																R/W-0h															

Table 12-82. ADCPPB1TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 1 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB1TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[15:13] will be ignored in 12 bit mode Reset type: SYSRSn

12.14.3.44 ADCPPB2CONFIG Register (Offset = 90h) [Reset = 00000001h]

ADCPPB2CONFIG is shown in Figure 12-82 and described in Table 12-83.

Return to the Summary Table.

ADC PPB{#} Config Register

Figure 12-82. ADCPPB2CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TWOSCOMPEN	RESERVED	CBCEN	RESERVED	CONFIG			
R/W-0h	R-0h	R/W-0h	R-0h	R/W-1h			

Table 12-83. ADCPPB2CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TWOSCOMPEN	R/W	0h	ADC Post Processing Block {#} Two's Complement Enable. When set this bit enables the post conversion hardware processing circuit that performs a two's complement on the output of the offset/reference subtraction unit before storing the result in the ADCPPB{#}RESULT register. 0 ADCPPB{#}RESULT = ADCRESULTx - ADCPPB{#}OFFREF 1 ADCPPB{#}RESULT = ADCPPB{#}OFFREF - ADCRESULTx Reset type: SYSRSn
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4	RESERVED	R	0h	Reserved

Table 12-83. ADCPPB2CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	CONFIG	R/W	1h	ADC Post Processing Block {#} Configuration. This bit field defines which SOC/EOC/RESULT is associated with this post processing block. 0000 SOC0/EOC0/RESULT0 is associated with post processing block {#} 0001 SOC1/EOC1/RESULT1 is associated with post processing block {#} 0010 SOC2/EOC2/RESULT2 is associated with post processing block {#} 0011 SOC3/EOC3/RESULT3 is associated with post processing block {#} 0100 SOC4/EOC4/RESULT4 is associated with post processing block {#} 0101 SOC5/EOC5/RESULT5 is associated with post processing block {#} 0110 SOC6/EOC6/RESULT6 is associated with post processing block {#} 0111 SOC7/EOC7/RESULT7 is associated with post processing block {#} 1000 SOC8/EOC8/RESULT8 is associated with post processing block {#} 1001 SOC9/EOC9/RESULT9 is associated with post processing block {#} 1010 SOC10/EOC10/RESULT10 is associated with post processing block {#} 1011 SOC11/EOC11/RESULT11 is associated with post processing block {#} 1100 SOC12/EOC12/RESULT12 is associated with post processing block {#} 1101 SOC13/EOC13/RESULT13 is associated with post processing block {#} 1110 SOC14/EOC14/RESULT14 is associated with post processing block {#} 1111 SOC15/EOC15/RESULT15 is associated with post processing block {#} Reset type: SYSRSn

12.14.3.45 ADCPPB2OFFCAL Register (Offset = 94h) [Reset = 00000000h]

ADCPPB2OFFCAL is shown in Figure 12-83 and described in Table 12-84.

Return to the Summary Table.

ADC PPB2 Offset Calibration Register

Figure 12-83. ADCPPB2OFFCAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										OFFCAL					
R-0h																										R/W-0h					

Table 12-84. ADCPPB2OFFCAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	OFFCAL	R/W	0h	ADC Post Processing Block 2 Offset Correction. This bit field can be used to digitally remove any system level offset inherent in the ADCIN circuit. This 6-bit signed value is subtracted from the ADC output before being stored in the ADCRESULT register. 00h No change. The ADC output is stored directly into ADCRESULT. 01h ADC output - 1 is stored into ADCRESULT. 02h ADC output - 2 is stored into ADCRESULT. ... 20h ADC output + 32 is stored into ADCRESULT. ... 3Fh ADC output + 1 is stored into ADCRESULT. NOTE: The subtraction will saturate at 0000h and 0FFFh before being stored into the ADCRESULT register. Note: in the case that multiple PPBs point to the same SOC, only the OFFCAL of the lowest numbered PPB will be applied. Reset type: SYSRSn

12.14.3.46 ADCPPB2OFFREF Register (Offset = 96h) [Reset = 00000000h]

ADCPPB2OFFREF is shown in Figure 12-84 and described in Table 12-85.

Return to the Summary Table.

ADC PPB2 Offset Reference Register

Figure 12-84. ADCPPB2OFFREF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				OFFREF											
R-0h																				R/W-0h											

Table 12-85. ADCPPB2OFFREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-0	OFFREF	R/W	0h	ADC Post Processing Block 2 Offset Correction. This bit field can be used to either calculate the feedback error or convert a unipolar signal to bipolar by subtracting a reference value. This 12-bit unsigned value is subtracted from the ADCRESULT register before being passed through an optional two's complement function and stored in the ADCPPB2RESULT register. This subtraction is not saturated. 0000h No change. The ADCRESULT value is passed on. 001h ADCRESULT - 1 is passed on. 002h ADCRESULT - 2 is passed on. ... 800h ADCRESULT - 2048 is passed on. ... FFFh ADCRESULT - 4096 is passed on. Reset type: SYSRSn

12.14.3.47 ADCPPB2TRIPHI Register (Offset = 98h) [Reset = 00000000h]

ADCPPB2TRIPHI is shown in Figure 12-85 and described in Table 12-86.

Return to the Summary Table.

ADC PPB2 Trip High Register

Figure 12-85. ADCPPB2TRIPHI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LIMITHI															
R-0h																R/W-0h															

Table 12-86. ADCPPB2TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITHI	R/W	0h	ADC Post Processing Block 2 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[15:13] will be ignored Reset type: SYSRSn

12.14.3.48 ADCPPB2TRIPLO Register (Offset = 9Ah) [Reset = 00000000h]

ADCPPB2TRIPLO is shown in Figure 12-86 and described in Table 12-87.

Return to the Summary Table.

ADC PPB2 Trip Low/Trigger Time Stamp Register

Figure 12-86. ADCPPB2TRIPLO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LIMITLO															
R-0h																R/W-0h															

Table 12-87. ADCPPB2TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 2 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB2TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[15:13] will be ignored in 12 bit mode Reset type: SYSRSn

12.14.3.49 ADCPPB3CONFIG Register (Offset = A0h) [Reset = 00000002h]

ADCPPB3CONFIG is shown in Figure 12-87 and described in Table 12-88.

Return to the Summary Table.

ADC PPB{#} Config Register

Figure 12-87. ADCPPB3CONFIG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TWOSCOMPEN	RESERVED	CBCEN	RESERVED	CONFIG			
R/W-0h	R-0h	R/W-0h	R-0h	R/W-2h			

Table 12-88. ADCPPB3CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	TWOSCOMPEN	R/W	0h	ADC Post Processing Block {#} Two's Complement Enable. When set this bit enables the post conversion hardware processing circuit that performs a two's complement on the output of the offset/reference subtraction unit before storing the result in the ADCPPB{#}RESULT register. 0 ADCPPB{#}RESULT = ADCRESULTx - ADCPPB{#}OFFREF 1 ADCPPB{#}RESULT = ADCPPB{#}OFFREF - ADCRESULTx Reset type: SYSRSn
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4	RESERVED	R	0h	Reserved

Table 12-88. ADCPPB3CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	CONFIG	R/W	2h	ADC Post Processing Block {#} Configuration. This bit field defines which SOC/EOC/RESULT is associated with this post processing block. 0000 SOC0/EOC0/RESULT0 is associated with post processing block {#} 0001 SOC1/EOC1/RESULT1 is associated with post processing block {#} 0010 SOC2/EOC2/RESULT2 is associated with post processing block {#} 0011 SOC3/EOC3/RESULT3 is associated with post processing block {#} 0100 SOC4/EOC4/RESULT4 is associated with post processing block {#} 0101 SOC5/EOC5/RESULT5 is associated with post processing block {#} 0110 SOC6/EOC6/RESULT6 is associated with post processing block {#} 0111 SOC7/EOC7/RESULT7 is associated with post processing block {#} 1000 SOC8/EOC8/RESULT8 is associated with post processing block {#} 1001 SOC9/EOC9/RESULT9 is associated with post processing block {#} 1010 SOC10/EOC10/RESULT10 is associated with post processing block {#} 1011 SOC11/EOC11/RESULT11 is associated with post processing block {#} 1100 SOC12/EOC12/RESULT12 is associated with post processing block {#} 1101 SOC13/EOC13/RESULT13 is associated with post processing block {#} 1110 SOC14/EOC14/RESULT14 is associated with post processing block {#} 1111 SOC15/EOC15/RESULT15 is associated with post processing block {#} Reset type: SYSRSn

12.14.3.50 ADCPPB3OFFCAL Register (Offset = A4h) [Reset = 00000000h]

ADCPPB3OFFCAL is shown in Figure 12-88 and described in Table 12-89.

Return to the Summary Table.

ADC PPB3 Offset Calibration Register

Figure 12-88. ADCPPB3OFFCAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										OFFCAL					
R-0h																										R/W-0h					

Table 12-89. ADCPPB3OFFCAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	OFFCAL	R/W	0h	ADC Post Processing Block 3 Offset Correction. This bit field can be used to digitally remove any system level offset inherent in the ADCIN circuit. This 6-bit signed value is subtracted from the ADC output before being stored in the ADCRESULT register. 00h No change. The ADC output is stored directly into ADCRESULT. 01h ADC output - 1 is stored into ADCRESULT. 02h ADC output - 2 is stored into ADCRESULT. ... 20h ADC output + 32 is stored into ADCRESULT. ... 3Fh ADC output + 1 is stored into ADCRESULT. NOTE: The subtraction will saturate at 0000h and 0FFFh before being stored into the ADCRESULT register. Note: in the case that multiple PPBs point to the same SOC, only the OFFCAL of the lowest numbered PPB will be applied. Reset type: SYSRSn

12.14.3.51 ADCPPB3OFFREF Register (Offset = A6h) [Reset = 00000000h]

ADCPPB3OFFREF is shown in Figure 12-89 and described in Table 12-90.

Return to the Summary Table.

ADC PPB3 Offset Reference Register

Figure 12-89. ADCPPB3OFFREF Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				OFFREF											
R-0h																				R/W-0h											

Table 12-90. ADCPPB3OFFREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-0	OFFREF	R/W	0h	ADC Post Processing Block 3 Offset Correction. This bit field can be used to either calculate the feedback error or convert a unipolar signal to bipolar by subtracting a reference value. This 12-bit unsigned value is subtracted from the ADCRESULT register before being passed through an optional two's complement function and stored in the ADCPPB3RESULT register. This subtraction is not saturated. 0000h No change. The ADCRESULT value is passed on. 001h ADCRESULT - 1 is passed on. 002h ADCRESULT - 2 is passed on. ... 800h ADCRESULT - 2048 is passed on. ... FFFh ADCRESULT - 4096 is passed on. Reset type: SYSRSn

12.14.3.52 ADCPPB3TRIPHI Register (Offset = A8h) [Reset = 00000000h]

ADCPPB3TRIPHI is shown in Figure 12-90 and described in Table 12-91.

Return to the Summary Table.

ADC PPB3 Trip High Register

Figure 12-90. ADCPPB3TRIPHI Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LIMITHI															
R-0h																R/W-0h															

Table 12-91. ADCPPB3TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITHI	R/W	0h	ADC Post Processing Block 3 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[15:13] will be ignored Reset type: SYSRSn

12.14.3.53 ADCPPB3TRIPLO Register (Offset = AAh) [Reset = 00000000h]

ADCPPB3TRIPLO is shown in Figure 12-91 and described in Table 12-92.

Return to the Summary Table.

ADC PPB3 Trip Low/Trigger Time Stamp Register

Figure 12-91. ADCPPB3TRIPLO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LIMITLO															
R-0h																R/W-0h															

Table 12-92. ADCPPB3TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 3 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB3TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[15:13] will be ignored in 12 bit mode Reset type: SYSRSn

12.14.3.54 ADCINTCYCLE Register (Offset = C0h) [Reset = 00000000h]

ADCINTCYCLE is shown in Figure 12-92 and described in Table 12-93.

Return to the Summary Table.

ADC Early Interrupt Generation Cycle

Figure 12-92. ADCINTCYCLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										DELAY					
R-0h																										R/W-0h					

Table 12-93. ADCINTCYCLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	DELAY	R/W	0h	ADC Early Interrupt Generation Cycle Delay: Defines the delay from the fall edge of ADCSOC in terms of system clock cycles, for the interrupt to be generated. Reset type: SYSRSn

12.14.3.55 ADCINLTRIM1 Register (Offset = C2h) [Reset = X0000000h]

ADCINLTRIM1 is shown in Figure 12-93 and described in Table 12-94.

Return to the Summary Table.

ADC Linearity Trim 1 Register

Figure 12-93. ADCINLTRIM1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM31TO0																															
R/W-Xh																															

Table 12-94. ADCINLTRIM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INLTRIM31TO0	R/W	Xh	ADC Linearity Trim Bits 31-0. This register should not be modified unless specifically indicated by TI Errata or other documentation. Modifying the contents of this register could cause this module to operate outside of datasheet specifications. Reset type: XRSn

12.14.3.56 ADCINLTRIM2 Register (Offset = C4h) [Reset = X0000000h]

ADCINLTRIM2 is shown in Figure 12-94 and described in Table 12-95.

Return to the Summary Table.

ADC Linearity Trim 2 Register

Figure 12-94. ADCINLTRIM2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM63TO32																															
R/W-Xh																															

Table 12-95. ADCINLTRIM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	INLTRIM63TO32	R/W	Xh	ADC Linearity Trim Bits 63-32. This register should not be modified unless specifically indicated by TI Errata or other documentation. Modifying the contents of this register could cause this module to operate outside of datasheet specifications. Reset type: XRSn

12.14.3.57 ADCREV2 Register (Offset = CEh) [Reset = 00000006h]

ADCREV2 is shown in Figure 12-95 and described in Table 12-96.

Return to the Summary Table.

ADC Wrapper Revision Register

Figure 12-95. ADCREV2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRAPPERREV															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRAPPERREV								WRAPPERTYPE							
R-0h								R-6h							

Table 12-96. ADCREV2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	WRAPPERREV	R	0h	ADC Revision. To allow documentation of differences between revisions. First version is labeled as 00h. Reset type: SYSRSn
7-0	WRAPPERTYPE	R	6h	ADC Wrapper Type. Always set to 6 for this ADC. Reset type: SYSRSn

12.14.3.58 REP1CTL Register (Offset = E0h) [Reset = 00000000h]

REP1CTL is shown in Figure 12-96 and described in Table 12-97.

Return to the Summary Table.

ADC Trigger Repeater 1 Control Register

Figure 12-96. REP1CTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
SWSYNC	RESERVED				SYNCINSEL		
R-0/W1S-0h	R-0h				R/W-0h		
15	14	13	12	11	10	9	8
RESERVED			TRIGGER				
R-0h			R/W-0h				
7	6	5	4	3	2	1	0
TRIGGEROVF	RESERVED	RESERVED	RESERVED	MODULEBUSY	RESERVED	RESERVED	RESERVED
R/W1C-0h	R/W1C-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R/W-0h

Table 12-97. REP1CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23	SWSYNC	R-0/W1S	0h	Trigger repeater 1 software force sync. On a sync. event, all registers in Repeater 1 are reset to a ready and waiting state. Value of NSEL is preserved. Note: SOCs associated with repeater 1 are not cleared. Reset type: SYSRSn
22-19	RESERVED	R	0h	Reserved
18-16	SYNCINSEL	R/W	0h	Trigger repeater 1 sync. input select. On a sync. event, all registers in Repeater 1 are reset to a ready and waiting state. Value of NSEL is preserved. Note: SOCs associated with repeater 1 are not cleared. 0h = Disable Syncin to Repeater 1 1h - 1Fh = Hardware syncin sources Reset type: SYSRSn
15-13	RESERVED	R	0h	Reserved
12-8	TRIGGER	R/W	0h	ADC Trigger Repeater 1 Trigger Select. Selects the trigger to modify via oversampling. 00h REPTRIG0 - Software only 01h-1FH Hardware triggers Reset type: SYSRSn

Table 12-97. REP1CTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	TRIGGEROVF	R/W1C	0h	ADC Trigger Repeater 1 Oversampled Trigger Overflow. Indicates that a trigger was dropped because a trigger arrived while the repeater was still generating repeated oversampled triggers (NCOUNT was not 0 or SOCs associated with Repeater 1 were still pending). Writing a 1 will clear this flag. Note: This flag won't be set when NSEL = 0 if a trigger arrives before the previous SOCs have completed, the trigger will be passed and the overflow flags of the SOCs that were still pending will be set. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will remain set. Reset type: SYSRSn
6	RESERVED	R/W1C	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	MODULEBUSY	R	0h	ADC Trigger Repeater 1 Module Busy Indicator. 0 = Repeater 1 is idle and can accept a new repeated trigger 1 = Repeater 1 still has repeated triggers remaining (NCOUNT > 0) or associated SOCs are still pending (SOCBUSY is 1) If a new oversampled trigger is received while the module is still busy, the TRIGGEROVF bit will be set and the trigger will be ignored. Reset type: SYSRSn
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	RESERVED	R/W	0h	Reserved

12.14.3.59 REP1N Register (Offset = E2h) [Reset = 00000000h]

REP1N is shown in Figure 12-97 and described in Table 12-98.

Return to the Summary Table.

ADC Trigger Repeater 1 N Select Register

Figure 12-97. REP1N Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED													NCOUNT		
R-0h													R-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														NSEL	
R-0h														R/W-0h	

Table 12-98. REP1N Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-16	NCOUNT	R	0h	ADC trigger repeater 1 trigger count. Indicates the number of triggers remaining to be generated. If a trigger is received corresponding to REP1CTL.TRIGSEL while NCOUNT is not 0 (the repeater is still busy generating the repeated triggers) then the trigger will be ignored and REP1CTL.TRIGOVF will be set to 1. Reset type: SYSRSn
15-2	RESERVED	R	0h	Reserved
1-0	NSEL	R/W	0h	ADC Trigger Repeater 1 selection of number of triggers. In oversampling mode, selects the number of repeated triggers. For each trigger received corresponding to REP1CTL.TRIGSEL, 2^{NSEL} triggers will be generated. 0 = 1 trigger is generated (pass-through) 1 = 2 triggers are generated 2 = 4 triggers are generated 3 = 8 triggers are generated In undersampling mode, selects the number triggers to be supressed. 1 out NSEL + 1 triggers received corresponding to REP1CTL.TRIGSEL will be passed through (the first trigger will be passed through and the subsequent NSEL triggers will be supressed). 0 = all triggers are passed 1 = 1 out of 2 triggers are passed 2 = 1 out of 3 triggers are passed ... 127 = 1 out of 128 triggers are passed Reset type: SYSRSn

12.14.3.60 REP1SPREAD Register (Offset = E6h) [Reset = 00000000h]

REP1SPREAD is shown in Figure 12-98 and described in Table 12-99.

Return to the Summary Table.

ADC Trigger Repeater 1 Spread Select Register

Figure 12-98. REP1SPREAD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPREADCOUNT																SPREAD															
R-0h																R/W-0h															

Table 12-99. REP1SPREAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SPREADCOUNT	R	0h	ADC trigger repeater 1 spread status. When a trigger is sent to the ADC in oversampling mode, this register will start counting down from SPREAD until SPREADCOUNT equals 0. The next repeated trigger to the ADC in oversampling mode will not occur until SPREADCOUNT is 0 (minimum time is complete) and REP1CTL.BUSY = 0 (SOCs associated with trigger repeater 1 are no longer pending). Reset type: SYSRSn
15-0	SPREAD	R/W	0h	ADC trigger repeater 1 spread delay configuration. In oversampling mode, defines the minimum number of SYSCLKs to wait before creating the next repeated trigger to the ADC. If SPREAD is less than the time needed for all SOCs associated with repeater 1 to sample and convert, then the repeater will generate triggers as fast as the ADC can convert the associated conversions. If SPREAD is greater than the time needed for all SOCs associated with repeater 1 to sample and convert, then repeated triggers to the ADC will be SPREAD SYSCLK cycles apart. 0 = oversampled repeated triggers occur as fast as the ADC can sample and convert associated SOCs 1 = time between repeated triggers is at least 1 SYSCLKs 2 = time between repeated triggers is at least 2 SYSCLKs ... 65535 = time between repeated triggers is at least 65535 SYSCLKs Reset type: SYSRSn

12.14.3.61 REP1FRC Register (Offset = E8h) [Reset = 00000000h]

REP1FRC is shown in Figure 12-99 and described in Table 12-100.

Return to the Summary Table.

ADC Trigger Repeater 1 Software Force Register

Figure 12-99. REP1FRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SWFRC
R-0h							R-0/W1S-0h

Table 12-100. REP1FRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SWFRC	R-0/W1S	0h	Write 1 to force a trigger to the Repeat Block 1 input regardless of the value of TRIGGER. Always reads 0. Reset type: SYSRSn

12.14.3.62 ADCPPB1LIMIT Register (Offset = 100h) [Reset = 00000000h]

ADCPPB1LIMIT is shown in Figure 12-100 and described in Table 12-101.

Return to the Summary Table.

ADC PPB1Conversion Count Limit Register

Figure 12-100. ADCPPB1LIMIT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LIMIT	
R-0h														R/W-0h	

Table 12-101. ADCPPB1LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	LIMIT	R/W	0h	Post Processing Block 1 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. 0 = No - accumulation 1 = 2 conversions are accumulated 2 = 4 conversions are accumulated 3 = 8 conversions are accumulated Reset type: SYSRSn

12.14.3.63 ADCPPBP1PCOUNT Register (Offset = 102h) [Reset = 00000000h]

ADCPPBP1PCOUNT is shown in Figure 12-101 and described in Table 12-102.

Return to the Summary Table.

ADC PPB1 Partial Conversion Count Register

Figure 12-101. ADCPPBP1PCOUNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													PCOUNT		
R-0h													R-0h		

Table 12-102. ADCPPBP1PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	PCOUNT	R	0h	Post Processing Block 1 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB1PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

12.14.3.64 ADCPPB1CONFIG2 Register (Offset = 104h) [Reset = 00000000h]

ADCPPB1CONFIG2 is shown in Figure 12-102 and described in Table 12-103.

Return to the Summary Table.

ADC PPB1 Sum Shift Register

Figure 12-102. ADCPPB1CONFIG2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
COMPSEL	RESERVED		OSINTSEL	SWSYNC	RESERVED		
R/W-0h	R-0h		R/W-0h	R-0/W1S-0h	R-0h		
7	6	5	4	3	2	1	0
RESERVED	SYNCINSEL			RESERVED	SHIFT		
R-0h	R/W-0h			R-0h	R/W-0h		

Table 12-103. ADCPPB1CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	COMPSEL	R/W	0h	Post Processing Block 1 Compare Source Select. This field determines whether ADCPPB1RESULT, ADCPPB1PSUM, or ADCPPB1SUM is used for the zero-crossing detect logic and threshold compare. 0 = ADCPPB1RESULT is used for compare logic 1 = ADCPPB1SUM is used for compare logic Note: when ADCPPB1PSUM is selected as the compare source and when a LIMIT match occurs (ADCPPB1LIMIT equals ADCPPB1COUNT) the ADCPPB1PSUM register will be cleared and the final sum will be loaded into ADCPPB1SUM. For this sample, the final sum, ADCPPB1SUM will be used for the comparison instead of ADCPPB1PSUM. Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved
12	OSINTSEL	R/W	0h	Post Processing Block 1 Interrupt Source Select. OSINT1 can be used to trigger an ADC interrupt (ADCINT1 through ADCINT4) via selection in the ADCINT1N2 or ADCINT3N4. This selection determines if a sync event can trigger OSINT1 in addition to a PCOUNT = LIMIT event. 0 = OSINT1 will be generated from PCOUNT = LIMIT only 1 = OSTIN1 will be generated form PCOUNT = LIMIT or a sync. event. Note: If a SYNC event would cause an OSINT one cycle after OSINT would have been cause by PCOUNT = LIMIT match, then the second OSINT is ignored. Reset type: SYSRSn
11	SWSYNC	R-0/W1S	0h	PPB 1 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn

Table 12-103. ADCPPB1CONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-7	RESERVED	R	0h	Reserved
6-4	SYNCINSEL	R/W	0h	PPB 1 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2-0	SHIFT	R/W	0h	Post Processing Block 1 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 7 : SUM = PSUM >> 7 Reset type: SYSRSn

12.14.3.65 ADCPPB1PSUM Register (Offset = 106h) [Reset = 00000000h]

ADCPPB1PSUM is shown in Figure 12-103 and described in Table 12-104.

Return to the Summary Table.

ADC PPB1 Partial Sum Register

Figure 12-103. ADCPPB1PSUM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGN																PSUM															
R-0h																R-0h															

Table 12-104. ADCPPB1PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	PSUM	R	0h	Post Processing Block 1 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB1RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be available 2 SYSCLK cycles after the associated ADCRESULT. Subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles. Reset type: SYSRSn

Chapter 13

Comparator Subsystem (CMPSS)

The Comparator Subsystem (CMPSS) consists of analog comparators and supporting circuits that are useful for power applications such as peak current mode control, switched-mode power, power factor correction, voltage trip monitoring, and so forth.

See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with modules of the same type, to determine the differences between the types, and for a list of device-specific differences within a type.

13.1 Introduction

The comparator subsystem is built around a number of modules. Each subsystem contains two comparators, two reference 12-bit DACs, and two digital filters. Comparators are denoted "H" or "L" within each module where “H” and “L” represent high and low, respectively. Each comparator generates a digital output which indicates whether the voltage on the positive input is greater than the voltage on the negative input. The positive input of the comparator is driven from an external pin or by the PGA (see the *Analog Subsystem* chapter for mux options available to the CMPSS). The negative input can be driven by an external pin or by the programmable reference 12-bit DAC . Each comparator output passes through a programmable digital filter that can remove spurious trip signals. An unfiltered output is also available if filtering is not required.

13.1.1 Features

Each CMPSS includes:

- Two analog comparators
- Two programmable reference 12-bit DACs (These DAC instances are 12-bit with a lower effective resolution as described in the data sheet)
- Two digital filters, max filter clock prescale = 2^{16}
- Ability to synchronize submodules with MCPWMSYNCPER
- Ability to synchronize output with SYSCLK
- Ability to latch output
- Ability to invert output
- Option to use hysteresis on the input
- Option for negative input of comparator to be driven by an external signal or by the reference DAC
- VDDA is the DAC reference voltage
- Option to use the low comparator DAC output on an external pin (select instances only, mutually exclusive with use of comparator functionality)

13.1.2 CMPSS Related Collateral

Foundational Materials

- C28x Academy - CMPSS
- *Real-Time Control Reference Guide*
  - Refer to the Comparator section

Expert Materials

- *Peak Current Mode Controlled PSFB Converter Reference Design Using C2000™ Real-time MCU*
- *Understanding and Applying Current-Mode Control Theory Application Report*

13.1.3 Block Diagram

The block diagram for the CMPSS is shown in Figure 13-1.

- CTRIPx (x= "H" or "L") signals are connected to the PWM X-BAR for MCPWM trip response. See the *Multi-Channel Pulse Width Modulator (MCPWM)* chapter for more details on the PWM X-BAR mux configuration.
- CTRIPxOUTx(x= "H" or "L") signals are connected to the Output X-BAR for external signaling. See the *General-Purpose Input/Output (GPIO)* chapter for more details on the Output X-BAR mux configuration.
- CMP3_LITE_DACL is the low comparator DAC output that is available only in the CMPSS3 module. To enable this DAC output, set the CMPSSCTL.CMP3LDACOUTEN register field in the analog subsystem.

Note

Enabling the CMP3_LITE_DACL to a pin disables the functionality to the associated COMPL in CMPSS. In this case, the inverting input of the COMPL must come from the CMPx_LN input pin. All other functions of the CMPSS module are retained.

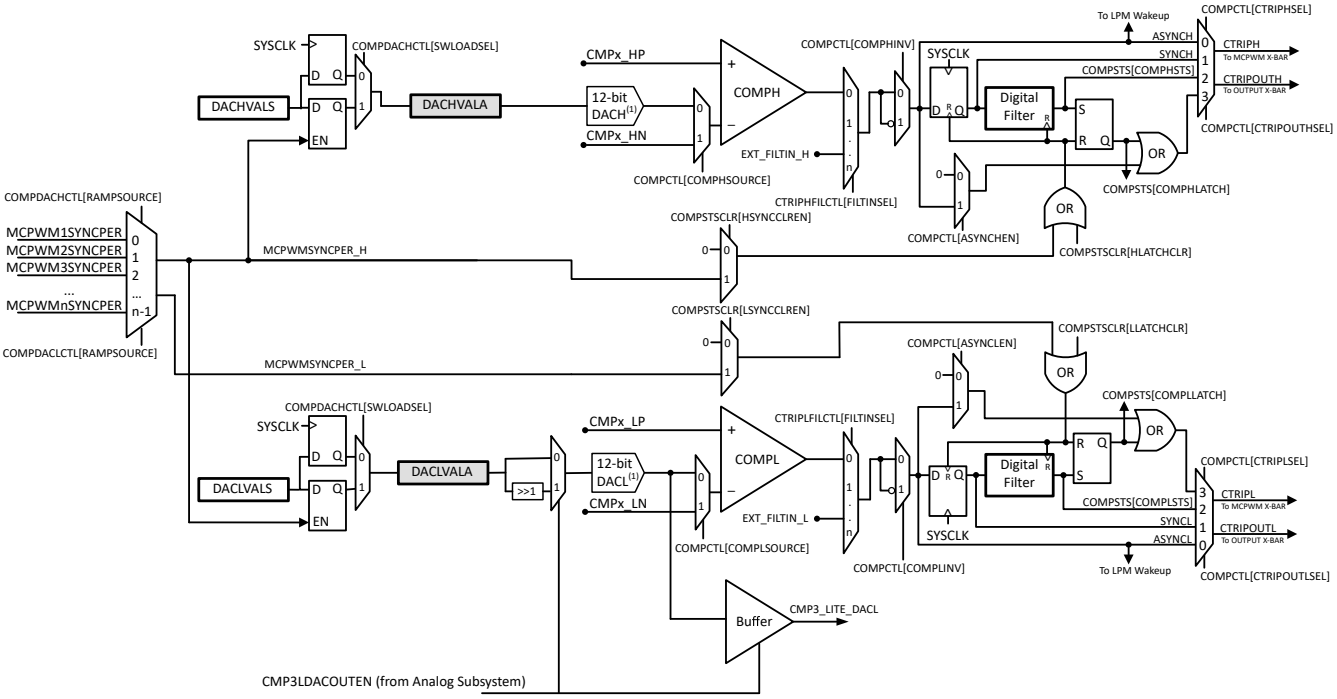


Figure 13-1. CMPSS_LITE Module Block Diagram

13.2 Comparator

The comparator generates a high digital output when the voltage on the positive input is greater than the voltage on the negative input, and a low digital output when the voltage on the positive input is less than the voltage on the negative input. The comparator is illustrated in Figure 13-2.

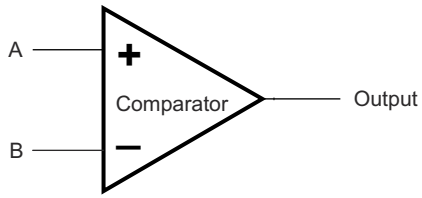


Figure 13-2. Comparator Block Diagram

Voltages	Output
Voltage A > Voltage B	1
Voltage A < Voltage B	0

13.3 Reference DAC

Each reference 12-bit DAC can be configured to drive a reference voltage into the negative input of the respective comparator. Some CMPSS instances also allow the low DAC output to be routed to a pin to act as an external DAC. In this case, all other CMPSS module functionality is not useable, including the high DAC, both comparators, and the digital filters.

Two sets of DACxVAL registers, DACxVALA and DACxVALS, are present for each reference 12-bit DAC. DACxVALA is a read-only register that actively controls the reference 12-bit DAC value. DACxVALS is a writable shadow register that loads into DACxVALA either immediately or synchronized with the next MCPWMSYNCPER event.

The operating range of the reference 12-bit DAC is bounded by DACREF and VSSA. The reference 12-bit DAC is illustrated in Figure 13-3.

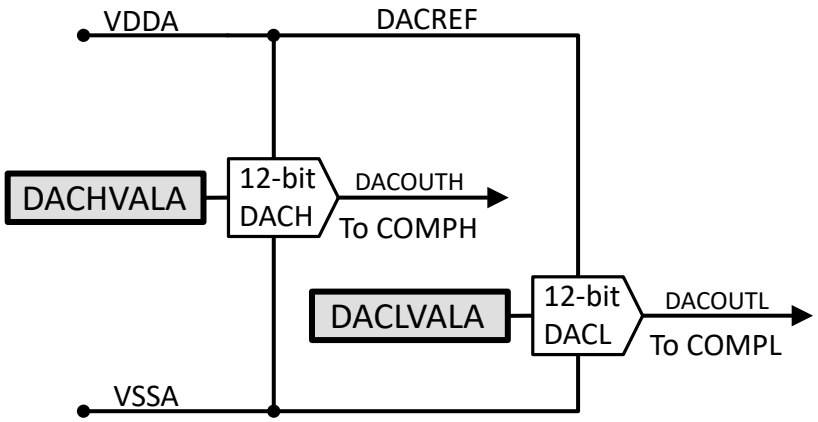


Figure 13-3. Reference DAC Block Diagram

The output of the reference 12-bit DAC can be calculated as:

$$DACOUT = \frac{(1 + DACVALA)*DACREF}{4096} \quad (10)$$

Note

- In the situations where both the DACH and DACL are driving the high and low comparators, a trip on one comparator can temporarily disturb the DAC output of the other comparator. The amount and length of time of this disturbance is specified in the device data sheet as “CMPSS DAC output disturbance” and “CMPSS DAC disturbance time”, respectively.

  Users must design the system carefully so that if the input signal crosses either DACH or DACL and trips the associated comparator, the input signal stays more than a “CMPSS DAC output disturbance” away from the other comparator trip point for “CMPSS DAC disturbance time”.
- The DACH setting must always be higher than the DACL setting. If the user is not using the DACL, the DACLVALS register must be set to 0 to avoid the comparator COMPL from tripping and affecting the DACH. In this case, there is no limitation on the DACHVALS setting. Accordingly, when not using the DACH, the user must set the DACHVALS register to the maximum.
- The CMPSS instance can be enabled before programming the reference DAC values.

13.4 Digital Filter

The digital filter works on a window of FIFO samples (SAMPWIN) taken from the input. The filter output resolves to the majority value of the sample window, where majority is defined by the threshold (THRESH) value. If the majority threshold is not satisfied, the filter output remains unchanged.

For proper operation, the value of THRESH must be greater than SAMPWIN / 2 and less than or equal to SAMPWIN.

A prescale function (CLKPRESCALE) determines the filter sampling rate, where the filter FIFO captures one sample every prescale system clocks. Old data from the FIFO is discarded.

Note that for SAMPWIN, THRESH and CLKPRESCALE, the internal number used by the digital filter is + 1 in all cases. In essence, samples = SAMPWIN + 1, threshold = THRESH + 1 and prescale = CLKPRESCALE + 1.

A conceptual model of the digital filter is shown in Figure 13-4.

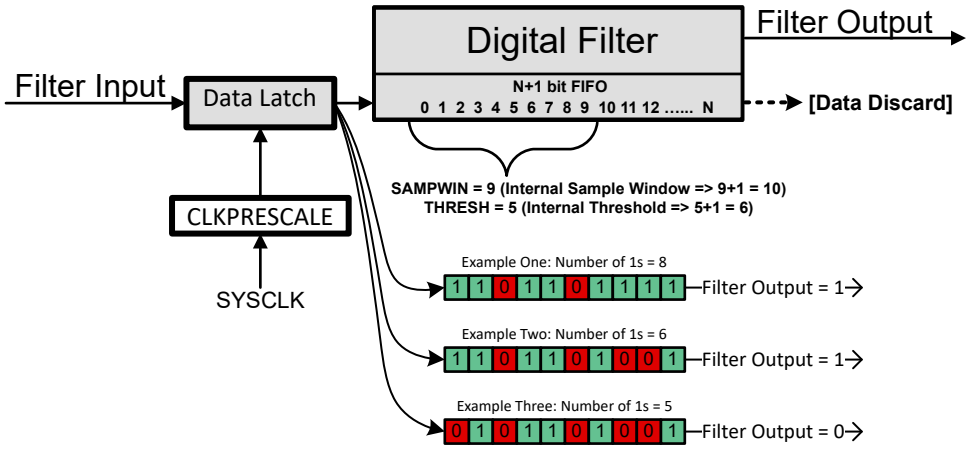


Figure 13-4. Digital Filter Behavior

Equivalent C code of the filter implementation is:

```
if (FILTER_OUTPUT == 0) {
    if (Num_1s_in_SAMPWIN >= THRESH) {
        FILTER_OUTPUT = 1;
    }
}
else {
    if (Num_0s_in_SAMPWIN >= THRESH) {
        FILTER_OUTPUT = 0;
    }
}
```

13.4.1 Filter Initialization Sequence

For proper operation of the digital filter, the following initialization sequence is recommended:

1. Configure and enable the comparator for operation.
2. Configure the digital filter parameters for operation:
   - Set SAMPWIN for the number of samples to monitor in the FIFO window.
   - Set THRESH for the threshold required for majority qualification.
   - Set CLKPRESCALE for the digital filter clock prescale value.
3. Initialize the sample values in the digital FIFO window by setting FILINIT.
4. Clear COMPSTS latch using COMPSTSCLR, if the latched path is desired.
5. Configure the CTRIP and CTRIPOUT signal paths.
6. If desired, configure the destination module, for example, MCPWM, GPIO, and so on to accept the filtered signals.

13.5 Using the CMPSS

13.5.1 LATCHCLR, and MCPWMSYNCPER Signals

The LATCHCLR signal holds the digital filter, synchronization block, and the latch output in reset (0) after the required delays. The LATCHCLR signal is activated in software using xLATCHCLR (x = H or L). The LATCHCLR signal can also be activated by MCPWMSYNCPER when xSYNCCLREN (x = H or L) is set. If a longer LATCHCLR signal is required, the EPWMBLANK signal can be used to extend the LATCHCLR signal by setting BLANKEN.

MCPWMxSYNCPER comes from the Time-Base submodule of the MCPWM, respectively. For a detailed description of how these two signals are generated, refer to the Time Base Counter Synchronization subsection in the *Multi-Channel Pulse Width Modulator (MCPWM)* chapter.

The MCPWMxSYNCPER signal that loads DACxVALA when COMPDACCTL[SWLOADSEL] = 1 is a level trigger load. If TBCTR and TBPRD of the MCPWM are both 0, MCPWMSYNCPER is held at level high and DACxVALA is loaded immediately from DACxVALS irrespective of the value of COMPDACCTL[SWLOADSEL]. Due to this, configure the MCPWM first before setting COMPDACCTL[SWLOADSEL] to 1.

Note

The name of the sync signal that the CMPSS receives from the MCPWM is referred to as MCPWMSYNCPER, PWMSYNCPER, and MCPWMxSYNCPER interchangeably in this document.

13.5.2 Synchronizer, Digital Filter, and Latch Delays

The synchronization block adds 1-2 SYSCLK cycle delays. If the digital filter is bypassed (all filter settings are 0), the digital filter adds 2 SYSCLK cycle delays. The latch adds 1 SYSCLK cycle delay.

13.5.3 Calibrating the CMPSS

The CMPSS has two sources of offset errors: comparator offset error and compdac offset error. In the data sheet, the comparator offset error is referred to as ***Input referred offset error*** and compdac offset error is referred to as ***Static offset error*** . See the device data sheet for the values.

If both inputs of the comparator are driven from a pin, only the comparator offset error applies. However if the inverting input of the comparator is driven from the compdac, then only the compdac offset error applies. This is because the compdac offset error includes comparator offset error.

Due to the offset errors, the CMPSS must be calibrated to make sure trips happen at the expected levels. The following flow outlines how the calibration can be performed if the inverting input of the comparator is driven from the compdac.

Notes before calibration:

1. A static DC signal is required on the non-inverting input of the comparator.
2. Hysteresis can be disabled for calibration and can be re-enabled after calibration is complete.
3. A noisy input can make calibration difficult, so use the latch with non-zero filter settings depending on how noisy is the signal on the non-inverting input.

This approach sweeps down the compdac:

1. Set the starting compdac value to max, 0xFFF.
   - Optional: Instead of setting the starting compdac value to maximum, set to ***Vtarget*** + ***Static offset error*** + ***Margin*** . Where ***Vtarget*** is the approximate DC voltage on the non-inverting input, ***Static offset error*** is the compdac offset error specification and ***Margin*** is some amount of guard band. This can lead to a faster calibration but only works if ***Vtarget*** is known. Alternatively, if ***Vtarget*** is unknown, the ADC can be used to convert ***Vtarget*** .
2. Decrement compdac value by 1.
3. Wait for compdac to settle.
4. Clear latch.
5. Wait for possible latch set.
6. If latch is set, trip code is found exit.
   - Optional: The trip code can be double checked by:
     a. Increasing compdac value by 1.
     b. Clear latch.
     c. Wait for possible latch set.
     d. Latch can be unset.
7. If latch is unset, go back to step 2 and repeat.

It is also possible to calibrate the CMPSS, if both inputs of the comparator are driven from a pin. For this case, the flow stays the same but the voltage on the inverting pin of the comparator is swept externally.

13.5.4 Enabling and Disabling the CMPSS Clock

If the clock to the CMPSS module is disabled while the comparator is active, the following behavior can be expected:

- The comparator remains unaffected and continues to trip from voltages on the inputs.
- If the reference 12-bit DAC is driving the negative input of the comparator, the voltage on the negative input remains static and unaffected but DACVALA can no longer be updated from the DACVALS.
- The synchronize block and digital filter freeze on the current states.

Enabling the clock to the CMPSS restores the clock to the state before the clock was disabled.

13.6 CMPSS DAC Output

Select instances of the CMPSS_LITE module can optionally route the low DAC output to an external pin to be used as an external DAC. This is subject to the following constraints:

- This is available only on select instances; consult the device data sheet to determine if the CMPSS DAC has been pinned out for a particular instance.
- In this case, the DAC output is not available to the COMPL inside the CMPSS module. The negative input to COMPL must be driven from the device pin in this case, if used. All other CMPSS module functionality, including the COMPH DAC source, is usable in this case.
- See the device data sheet for effective resolution of the CMPSS DAC output.

Note

The CMPSS low DAC output does not necessarily appear on the same pins that are available for low comparator input. Consult the pinout of the device data sheet to determine which pin the CMPSS output DAC is available on.

13.7 Software

13.7.1 CMPSS Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/cmpss

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

13.7.2 CMPSS_LITE Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/cmpss_lite

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

13.7.2.1 CMPSSLITE Asynchronous Trip

FILE: cmpss_lite_ex1_asynch.c

This example enables the CMPSSLITE2 COMPH comparator and feeds the asynchronous CTRIPOUTH signal to the GPIO4/OUTPUTXBAR3 pin and CTRIPH to GPIO0/MCPWM1A.

CMPSS is configured to generate trip signals to trip the MCPWM signals. CMPIN2P is used to give positive input and internal DAC is configured to provide the negative input. Internal DAC is configured to provide a signal at VDD/2. An MCPWM signal is generated at GPIO0 and is configured to be tripped by CTRIPOUTH.

When a low input(VSS) is provided to CMPIN2P,

- Trip signal(GPIO4) output is low
- PWM1A(GPIO0) gives a PWM signal

When a high input(higher than VDD/2) is provided to CMPIN2P,

- Trip signal(GPIO4) output turns high
- PWM1A(GPIO0) gets tripped and outputs as high

PWMXBAR is used to provide the trip signal (CTRIPH) to MCPWM. The trip signal is inverted in PWMXBAR since MCPWM trip zone signals are active low signals. *External Connections*

- Give input on CMPIN2P. The pin is shared with ADCINA12 for CMPHPMXSEL = 1
- Outputs can be observed on GPIO4 and GPIO0 using an oscilloscope

Watch Variables

- None

13.8 CMPSS Registers

This Section describes the CMPSS Registers.

13.8.1 CMPSS Base Address Table

Table 13-1. CMPSS Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
CmpssLite1Regs	CMPSS_LITE_REGS	CMPSSLITE1_BASE	0x0000_5500	YES	YES
CmpssLite2Regs	CMPSS_LITE_REGS	CMPSSLITE2_BASE	0x0000_5540	YES	YES
CmpssLite3Regs	CMPSS_LITE_REGS	CMPSSLITE3_BASE	0x0000_5580	YES	YES

13.8.2 CMPSS_LITE_REGS Registers

Table 13-2 lists the memory-mapped registers for the CMPSS_LITE_REGS registers. All register offset addresses not listed in Table 13-2 should be considered as reserved locations and the register contents should not be modified.

Table 13-2. CMPSS_LITE_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	COMPCTL	CMPSS Comparator Control Register	EALLOW
1h	COMPHYSCTL	CMPSS Comparator Hysteresis Control Register	EALLOW
2h	COMPSTS	CMPSS Comparator Status Register	
3h	COMPSTSCLR	CMPSS Comparator Status Clear Register	EALLOW
4h	COMPDACHCTL	CMPSS High DAC Control Register	EALLOW
6h	DACHVALS	CMPSS High DAC Value Shadow Register	
7h	DACHVALA	CMPSS High DAC Value Active Register	
12h	DACLVALS	CMPSS Low DAC Value Shadow Register	
13h	DACLVALA	CMPSS Low DAC Value Active Register	
16h	CTRIPLFILCTL	CTRIPL Filter Control Register	EALLOW
17h	CTRIPLFILCLKCTL	CTRIPL Filter Clock Control Register	EALLOW
18h	CTRIPHFILCTL	CTRIPH Filter Control Register	EALLOW
19h	CTRIPHFILCLKCTL	CTRIPH Filter Clock Control Register	EALLOW
1Ah	COMPLOCK	CMPSS Lock Register	EALLOW
24h	COMPDACLCTL	CMPSS Low DAC Control Register	EALLOW
37h	CTRIPLFILCLKCTL2	CTRIPL Filter Clock Control Register 2	EALLOW
39h	CTRIPHFILCLKCTL2	CTRIPH Filter Clock Control Register 2	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 13-3 shows the codes that are used for access types in this section.

Table 13-3. CMPSS_LITE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value

13.8.2.1 COMPCTL Register (Offset = 0h) [Reset = 0000h]

COMPCTL is shown in Figure 13-5 and described in Table 13-4.

Return to the Summary Table.

CMPSS Comparator Control Register

Figure 13-5. COMPCTL Register

15	14	13	12	11	10	9	8
COMPDACE	ASYNCLEN	CTRIPOUTLSEL		CTRIPLSEL		COMPLINV	COMPLSOURCE
R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	ASYNCHEN	CTRIPOUTHSEL		CTRIPHSEL		COMPHINV	COMPHSOURCE
R-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h

Table 13-4. COMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	COMPDACE	R/W	0h	Comparator/DAC enable. 0 Comparator/DAC disabled 1 Comparator/DAC enabled Reset type: SYSRSn
14	ASYNCLEN	R/W	0h	Low comparator asynchronous path enable. Allows asynchronous comparator output to feed into OR gate with latched digital filter signal when CTRIPLSEL=3 or CTRIPOUTLSEL=3. 0 Asynchronous comparator output does not feed into OR gate with latched digital filter output 1 Asynchronous comparator output feeds into OR gate with latched digital filter output Reset type: SYSRSn
13-12	CTRIPOUTLSEL	R/W	0h	Low comparator CTRIPOUTL source select. 0 Asynchronous comparator output drives CTRIPOUTL 1 Synchronous comparator output drives CTRIPOUTL 2 Output of digital filter drives CTRIPOUTL 3 Latched output of digital filter drives CTRIPOUTL Reset type: SYSRSn
11-10	CTRIPLSEL	R/W	0h	Low comparator CTRIPL source select. 0 Asynchronous comparator output drives CTRIPL 1 Synchronous comparator output drives CTRIPL 2 Output of digital filter drives CTRIPL 3 Latched output of digital filter drives CTRIPL Reset type: SYSRSn
9	COMPLINV	R/W	0h	Low comparator output invert. 0 Output of comparator is not inverted 1 Output of comparator is inverted Reset type: SYSRSn
8	COMPLSOURCE	R/W	0h	Low comparator input source. 0 Inverting input of comparator driven by internal DAC 1 Inverting input of comparator driven through external pin Reset type: SYSRSn
7	RESERVED	R	0h	Reserved

Table 13-4. COMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	ASYNCHEN	R/W	0h	High comparator asynchronous path enable. Allows asynchronous comparator output to feed into OR gate with latched digital filter signal when CTRIPHSEL=3 or CTRIPOUTHSEL=3. 0 Asynchronous comparator output does not feed into OR gate with latched digital filter output 1 Asynchronous comparator output feeds into OR gate with latched digital filter output Reset type: SYSRSn
5-4	CTRIPOUTHSEL	R/W	0h	High comparator CTRIPOUTH source select. 0 Asynchronous comparator output drives CTRIPOUTH 1 Synchronous comparator output drives CTRIPOUTH 2 Output of digital filter drives CTRIPOUTH 3 Latched output of digital filter drives CTRIPOUTH Reset type: SYSRSn
3-2	CTRIPHSEL	R/W	0h	High comparator CTRIPH source select. 0 Asynchronous comparator output drives CTRIPH 1 Synchronous comparator output drives CTRIPH 2 Output of digital filter drives CTRIPH 3 Latched output of digital filter drives CTRIPH Reset type: SYSRSn
1	COMPHINV	R/W	0h	High comparator output invert. 0 Output of comparator is not inverted 1 Output of comparator is inverted Reset type: SYSRSn
0	COMPHSOURCE	R/W	0h	High comparator input source. 0 Inverting input of comparator driven by internal DAC 1 Inverting input of comparator driven through external pin Reset type: SYSRSn

13.8.2.2 COMPHYSCTL Register (Offset = 1h) [Reset = 0000h]

COMPHYSCTL is shown in Figure 13-6 and described in Table 13-5.

Return to the Summary Table.

CMPSS Comparator Hysteresis Control Register

Figure 13-6. COMPHYSCTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				COMPHYS			
R-0h				R/W-0h			

Table 13-5. COMPHYSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	COMPHYS	R/W	0h	Comparator hysteresis. Sets the amount of hysteresis on the comparator inputs. 0 None 1 Set to typical hysteresis 2 Set to 2x of typical hysteresis 3 Set to 3x of typical hysteresis 4 Set to 4x of typical hysteresis others : undefined Reset type: SYSRSn

13.8.2.3 COMPSTS Register (Offset = 2h) [Reset = 0000h]

COMPSTS is shown in Figure 13-7 and described in Table 13-6.

Return to the Summary Table.

CMPSS Comparator Status Register

Figure 13-7. COMPSTS Register

15	14	13	12	11	10	9	8
RESERVED						COMPLLATCH	COMPLSTS
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED						COMPHLATCH	COMPHSTS
R-0h						R-0h	R-0h

Table 13-6. COMPSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	COMPLLATCH	R	0h	Latched value of low comparator digital filter output Reset type: SYSRSn
8	COMPLSTS	R	0h	Low comparator digital filter output Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1	COMPHLATCH	R	0h	Latched value of high comparator digital filter output Reset type: SYSRSn
0	COMPHSTS	R	0h	High comparator digital filter output Reset type: SYSRSn

13.8.2.4 COMPSTSCLR Register (Offset = 3h) [Reset = 0000h]

COMPSTSCLR is shown in Figure 13-8 and described in Table 13-7.

Return to the Summary Table.

CMPSS Comparator Status Clear Register

Figure 13-8. COMPSTSCLR Register

15	14	13	12	11	10	9	8
RESERVED					LSYNCCLREN	LLATCHCLR	RESERVED
R-0h					R/W-0h	R-0/W1S-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED					HSYNCCLREN	HLATCHCLR	RESERVED
R-0h					R/W-0h	R-0/W1S-0h	R-0h

Table 13-7. COMPSTSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R	0h	Reserved
10	LSYNCCLREN	R/W	0h	Low comparator latch PWMSYNCPER clear. Enable PWMSYNCPER reset of low comparator digital filter output latch COMPSTS[COMPLLATCH]. 0 PWMSYNCPER will not reset latch 1 PWMSYNCPER will reset latch Reset type: SYSRSn
9	LLATCHCLR	R-0/W1S	0h	Low comparator latch software clear. Perform software reset of low comparator digital filter output latch COMPSTS[COMPLLATCH]. Reads always return 0. 0 No effect 1 Generate a single pulse of latch reset signal for COMPSTS[COMPLLATCH] Reset type: SYSRSn
8-3	RESERVED	R	0h	Reserved
2	HSYNCCLREN	R/W	0h	High comparator latch PWMSYNCPER clear. Enable PWMSYNCPER reset of high comparator digital filter output latch COMPSTS[COMPHLATCH]. 0 PWMSYNCPER will not reset latch 1 PWMSYNCPER will reset latch Reset type: SYSRSn
1	HLATCHCLR	R-0/W1S	0h	High comparator latch software clear. Perform software reset of high comparator digital filter output latch COMPSTS[COMPHLATCH]. Reads always return 0. 0 No effect 1 Generate a single pulse of latch reset signal for COMPSTS[COMPHLATCH] Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

13.8.2.5 COMPDACHCTL Register (Offset = 4h) [Reset = 0000h]

COMPDACHCTL is shown in Figure 13-9 and described in Table 13-8.

Return to the Summary Table.

CMPSS High DAC Control Register

Figure 13-9. COMPDACHCTL Register

15	14	13	12	11	10	9	8
RESERVED		RESERVED	RESERVED	RESERVED			
R/W-0h		R/W-0h	R/W-0h	R/W-0h			
7	6	5	4	3	2	1	0
SWLOADSEL	RESERVED	RESERVED	RAMPSOURCE				RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h				R/W-0h

Table 13-8. COMPDACHCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11-8	RESERVED	R/W	0h	Reserved
7	SWLOADSEL	R/W	0h	Software load select. Determines whether DACxVALA is updated from DACxVALS on SYSCLK or MCPWMSYNCPER. 0 DACxVALA is updated from DACxVALS on SYSCLK 1 DACxVALA is updated from DACxVALS on MCPWMSYNCPER Reset type: SYSRSn
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4-1	RAMPSOURCE	R/W	0h	MCPWMSYNCPER source select. Determines which MCPWMnSYNCPER signal is used within the COMPH Where n represents the maximum number of MCPWMSYNCPER signals available on the device: 0 MCPWM1SYNCPER 1 MCPWM2SYNCPER 2 MCPWM3SYNCPER ... n-1 MCPWMnSYNCPER Reset type: SYSRSn
0	RESERVED	R/W	0h	Reserved

13.8.2.6 DACHVALS Register (Offset = 6h) [Reset = 0000h]

DACHVALS is shown in Figure 13-10 and described in Table 13-9.

Return to the Summary Table.

CMPSS High DAC Value Shadow Register

Figure 13-10. DACHVALS Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DACVAL							
R/W-0h							

Table 13-9. DACHVALS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R/W	0h	High DAC shadow value. When COMPDACCTL[DACSOURCE]=0, the value of DACHVALS is loaded into DACHVALA on the trigger signal selected by COMPDACCTL[SWLOADSEL]. Reset type: SYSRSn

13.8.2.7 DACHVALA Register (Offset = 7h) [Reset = 0000h]

DACHVALA is shown in Figure 13-11 and described in Table 13-10.

Return to the Summary Table.

CMPSS High DAC Value Active Register

Figure 13-11. DACHVALA Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DACVAL							
R-0h							

Table 13-10. DACHVALA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R	0h	High DAC active value. Value that is actively driven by the high DAC. Reset type: SYSRSn

13.8.2.8 DACLVALS Register (Offset = 12h) [Reset = 0000h]

DACLVALS is shown in Figure 13-12 and described in Table 13-11.

Return to the Summary Table.

CMPSS Low DAC Value Shadow Register

Figure 13-12. DACLVALS Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DACVAL							
R/W-0h							

Table 13-11. DACLVALS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R/W	0h	Low DAC shadow value. value to be loaded into DACLVALA on the trigger signal selected by COMPDACCTL[SWLOADSEL]. Reset type: SYSRSn

13.8.2.9 DACLVALA Register (Offset = 13h) [Reset = 0000h]

DACLVALA is shown in Figure 13-13 and described in Table 13-12.

Return to the Summary Table.

CMPSS Low DAC Value Active Register

Figure 13-13. DACLVALA Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DACVAL							
R-0h							

Table 13-12. DACLVALA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R	0h	Low DAC active value. Value that is actively driven by the low DAC. Reset type: SYSRSn

13.8.2.10 CTRIPLFILCTL Register (Offset = 16h) [Reset = 0000h]

CTRIPLFILCTL is shown in Figure 13-14 and described in Table 13-13.

Return to the Summary Table.

CTRIPL Filter Control Register

Figure 13-14. CTRIPLFILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN						FILTINSEL	
R/W-0h						R/W-0h	

Table 13-13. CTRIPLFILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Low filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14-9	THRESH	R/W	0h	Low filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	Low filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
2-0	FILTINSEL	R/W	0h	Low filter Input Mux Select Bit 0 Selects the COMPL output as the filter input 1 Selects the external signal EXT_FILTIN_L[1] as the filter input 2 Selects the external signal EXT_FILTIN_L[2] as the filter input ... ... 7 Selects the external signal EXT_FILTIN_L[7] as the filter input Reset type: SYSRSn

13.8.2.11 CTRIPLFILCLKCTL Register (Offset = 17h) [Reset = 0000h]

CTRIPLFILCLKCTL is shown in Figure 13-15 and described in Table 13-14.

Return to the Summary Table.

CTRIPL Filter Clock Control Register

Figure 13-15. CTRIPLFILCLKCTL Register

15	14	13	12	11	10	9	8
CLKPRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 13-14. CTRIPLFILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CLKPRESCALE	R/W	0h	Low filter sample clock prescale. Number of system clocks between samples is CLKPRESCALE+1. Reset type: SYSRSn

13.8.2.12 CTRIPHFILCTL Register (Offset = 18h) [Reset = 0000h]

CTRIPHFILCTL is shown in Figure 13-16 and described in Table 13-15.

Return to the Summary Table.

CTRIPH Filter Control Register

Figure 13-16. CTRIPHFILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN						FILTINSEL	
R/W-0h						R/W-0h	

Table 13-15. CTRIPHFILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	High filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14-9	THRESH	R/W	0h	High filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	High filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
2-0	FILTINSEL	R/W	0h	High filter Input Mux Select Bit 0 Selects the COMPL output as the filter input 1 Selects the external signal EXT_FILTIN_H[1] as the filter input 2 Selects the external signal EXT_FILTIN_H[2] as the filter input ... ... 7 Selects the external signal EXT_FILTIN_H[7] as the filter input Reset type: SYSRSn

13.8.2.13 CTRIPHFILCLKCTL Register (Offset = 19h) [Reset = 0000h]

CTRIPHFILCLKCTL is shown in Figure 13-17 and described in Table 13-16.

Return to the Summary Table.

CTRIPH Filter Clock Control Register

Figure 13-17. CTRIPHFILCLKCTL Register

15	14	13	12	11	10	9	8
CLKPRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 13-16. CTRIPHFILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CLKPRESCALE	R/W	0h	High filter sample clock prescale. Number of system clocks between samples is CLKPRESCALE+1. Reset type: SYSRSn

13.8.2.14 COMPLOCK Register (Offset = 1Ah) [Reset = 0000h]

COMPLOCK is shown in Figure 13-18 and described in Table 13-17.

Return to the Summary Table.

CMPSS Lock Register

Figure 13-18. COMPLOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	CTRIP	DACCTL	COMPHYSCTL	COMPCTL
R-0h			R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 13-17. COMPLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4	RESERVED	R/WSonce	0h	Reserved
3	CTRIP	R/WSonce	0h	Lock write-access to the CTRIPxFILTCTL and CTRIPxFILCLKCTL* registers. 0 CTRIPxFILCTL and CTRIPxFILCLKCTL* registers are not locked. Write 0 to this bit has no effect. 1 CTRIPxFILCTL and CTRIPxFILCLKCTL* registers are locked. Only a system reset can clear this bit. Reset type: SYSRSn
2	DACCTL	R/WSonce	0h	Lock write-access to the COMPDAC*CTL* register(s). 0 COMPDAC*CTL* register(s) not locked. Write 0 to this bit has no effect. 1 COMPDAC*CTL* register(s) locked. Only a system reset can clear this bit. Reset type: SYSRSn
1	COMPHYSCTL	R/WSonce	0h	Lock write-access to the COMPHYSCTL register. 0 COMPHYSCTL register is not locked. Write 0 to this bit has no effect. 1 COMPHYSCTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn
0	COMPCTL	R/WSonce	0h	Lock write-access to the COMPCTL register. 0 COMPCTL register is not locked. Write 0 to this bit has no effect. 1 COMPCTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn

13.8.2.15 COMPDACLCTL Register (Offset = 24h) [Reset = 0000h]

COMPDACLCTL is shown in Figure 13-19 and described in Table 13-18.

Return to the Summary Table.

CMPSS Low DAC Control Register

Figure 13-19. COMPDACLCTL Register

15	14	13	12	11	10	9	8
RESERVED		RESERVED	RESERVED	RESERVED			
R-0h		R/W-0h	R/W-0h	R/W-0h			
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RAMPSOURCE				RESERVED
R-0h	R/W-0h	R-0h	R/W-0h				R/W-0h

Table 13-18. COMPDACLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11-8	RESERVED	R/W	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R	0h	Reserved
4-1	RAMPSOURCE	R/W	0h	MCPWMSYNCPER source select. Determines which MCPWMnSYNCPER signal is used within the COMPL Where n represents the maximum number of MCPWMSYNCPER signals available on the device: 0 MCPWM1SYNCPER 1 MCPWM2SYNCPER 2 MCPWM3SYNCPER ... n-1 MCPWMnSYNCPER Reset type: SYSRSn
0	RESERVED	R/W	0h	Reserved

13.8.2.16 CTRIPLFILCLKCTL2 Register (Offset = 37h) [Reset = 0000h]

CTRIPLFILCLKCTL2 is shown in Figure 13-20 and described in Table 13-19.

Return to the Summary Table.

CTRIPL Filter Clock Control Register 2

Figure 13-20. CTRIPLFILCLKCTL2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CLKPRESCALEU							
R/W-0h							

Table 13-19. CTRIPLFILCLKCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	CLKPRESCALEU	R/W	0h	COMP Low filter sample clock prescale Upper Bits. The effective prescale value is (CLKPRESCALEH:CLKPRESCALE)+1 Reset type: SYSRSn

13.8.2.17 CTRIPHFILCLKCTL2 Register (Offset = 39h) [Reset = 0000h]

CTRIPHFILCLKCTL2 is shown in Figure 13-21 and described in Table 13-20.

Return to the Summary Table.

CTRIPH Filter Clock Control Register 2

Figure 13-21. CTRIPHFILCLKCTL2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CLKPRESCALEU							
R/W-0h							

Table 13-20. CTRIPHFILCLKCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	CLKPRESCALEU	R/W	0h	COMP High filter sample clock prescale Upper Bits. The effective prescale value is (CLKPRESCALEH:CLKPRESCALE)+1 Reset type: SYSRSn

Chapter 14

Programmable Gain Amplifier (PGA)

The Programmable Gain Amplifier (PGA) is used to amplify an input voltage for the purpose of increasing the dynamic range of the downstream ADC and CMPSS modules.

14.1 Programmable Gain Amplifier (PGA) Overview

The integrated PGA helps to reduce cost and design effort for many control applications that traditionally require external, standalone amplifiers. On-chip integration makes sure that the PGA is compatible with the downstream analog-to-digital converter (ADC) and comparator subsystem (CMPSS) modules. Software selectable gain, filter settings, and different operational modes make the PGA adaptable to various performance needs.

14.1.1 Features

Features available to PGA modules are:

- Rail to rail input and output voltage within VDDA and VSSA range
- Programmable gain modes including unity gain and other values from x2 to x64
- Standalone gain mode using off-chip passive components
- Post-gain filtering using on-chip resistors
- Differential input support
- Hardware assisted chopping for offset reduction
- Support for Kelvin ground connections using PGA_INM pins
- 3-to-1 mux on positive input pins

14.1.2 Block Diagram

Figure 14-1 shows the block diagram of the PGA. The active component in the PGA is an embedded operational amplifier (op-amp) that is configurable as a non-inverting or inverting amplifier with internal feedback resistors. These internal feedback resistor values are paired to produce software selectable voltage gains.

Six PGA signals are available at the device pins:

- PGA_INP1, PGA_INP2, and PGA_INP3 are multiplexed to get the positive input to the PGA op-amp.
- PGA_INM1 and PGA_INM2 are multiplexed to get the negative input to the PGA op-amp.
- PGA_OUT supports op-amp output filtering with RC components. The filtered signal is available for sampling and monitoring by on-chip ADC and CMPSS modules.

Take note that PGA_OUT _INT is an internal signal at the op-amp output, which is available for sampling and monitoring by the internal ADC and CMPSS modules.

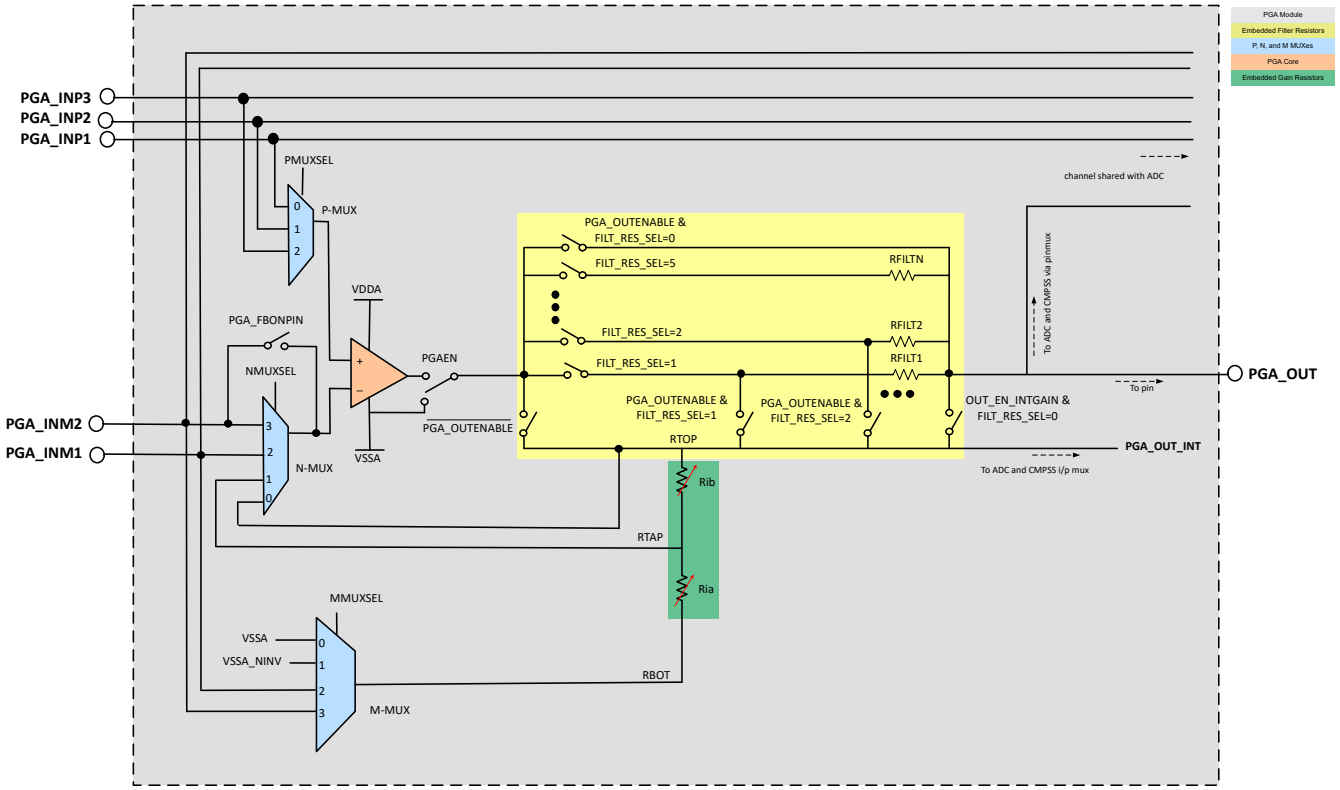


Figure 14-1. PGA Block Diagram

14.2 Linear Output Range

The absolute output range of the PGA is bounded by the analog VDDA and VSSA supplies – the PGA cannot produce output voltages greater than VDDA or less than VSSA.

Although the PGA can produce full-scale output across the absolute voltage range of VSSA to VDDA, the amplifier output is only linear within a subset of the absolute range. This reduced range is referred to as the linear output range.

The PGA performance specifications in the device data sheet only apply to the linear output range. For best performance, the input signal can be conditioned in such a way that the PGA stays within the linear output range during normal system operation.

Note

The voltage input range required to operate the PGA in the linear output range is unique for each gain mode. See the device data sheet for the linear output range.

14.3 Gain Values

Gain values of PGA are software-selectable using the PGACTL[GAIN] register field. The gain of the PGA in the subtractor and non-inverting modes is determined by a preset ratio between resistors R_{ia} and R_{ib}.

The target values for the gain resistors for subtractor and non-inverting modes are shown in Table 14-1.

Table 14-1. Different Gain Values and Corresponding Resistor Values

PGACTL[GAIN]	Non-inverting Gain	Inverting Gain	R_{ia}	R_{ib}
000	1	NA	256K	0(1)
001	2	-1	16K	16K
010	4	-3	8K	24K
011	8	-7	8K	56K
100	16	-15	8K	120K
101	32	-31	8K	248K
110	64	-63	4K	252K

(1) These values are nominal; tolerance is specified in electrical spec table given in the device data sheet.

Note

Changing the gain mode during normal operation is allowed, but a minimum configuration settling time can be observed when doing so. See the device data sheet for the gain switch settling time.

14.4 Modes of Operation

PGA can support four different operational modes based on the values in MUXSEL register, which are given in Table 14-2:

1. Buffer mode: PGA works in unity gain mode.
2. Standalone mode: PGA operates as a conventional op-amp.
3. Non-inverting mode: PGA works as a non-inverting op-amp.
4. Subtractor mode: PGA output voltage equals to the subtraction of the two inputs.

Table 14-2. Modes of Operation

Mode	MUXSEL[PMUXSEL]	MUXSEL[NMUXSEL]	MUXSEL[MMUXSEL]
Buffer mode	0	0	0
Standalone mode	0	2	0
Non-inverting mode	0	1	1
Subtractor mode	0	1	2 or 3

14.4.1 Buffer Mode

In this mode, PGA provides a high input impedance, low output impedance, and a voltage gain of approximately one. The block diagram of PGA in this mode is shown in Figure 14-2. This mode is also known as voltage follower, or unity gain mode. Even though this mode does not provide any voltage amplification, this mode is extremely useful because this mode prevents one stage's input impedance from loading the prior stage's output impedance. Not only does buffer mode isolate the load from the signal source, but also offers impedance matching between different parts of a circuit. The buffer op-amp also has a low output impedance that can drive a load without being affected by the load's impedance. Moreover, the low output impedance helps to maintain the signal voltage across the load, preventing voltage drops that can affect the performance of the circuit.

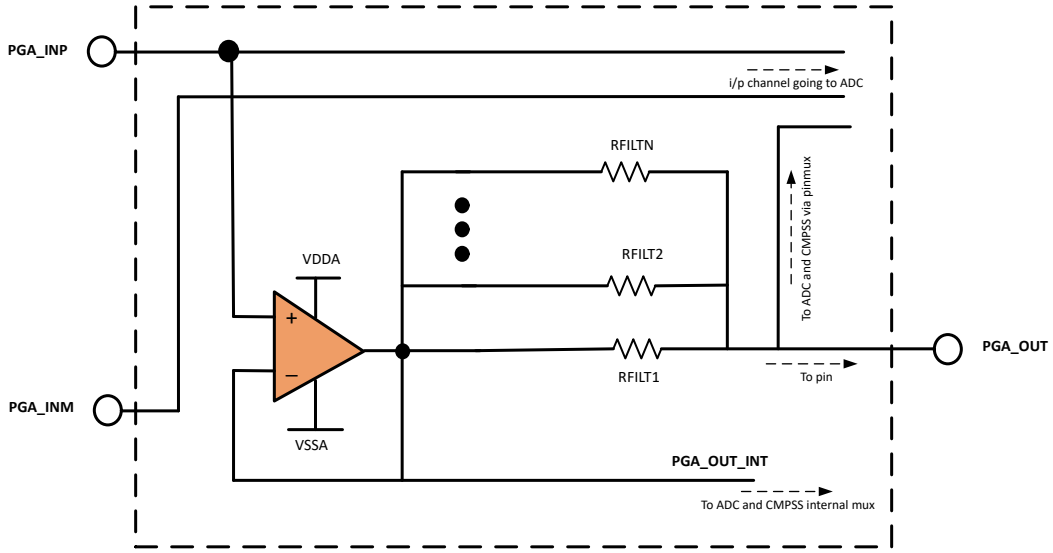


Figure 14-2. Buffer Mode

14.4.2 Standalone Mode

In this mode, a standalone operational amplifier is available that can support a number of analog applications using a minimum number of external components. Figure 14-3 Shows the block diagram of PGA in this mode. In this mode, the output voltage is determined by the external resistors connected to the inverting and non-inverting pins, as well as the feedback circuitry connected to the output.

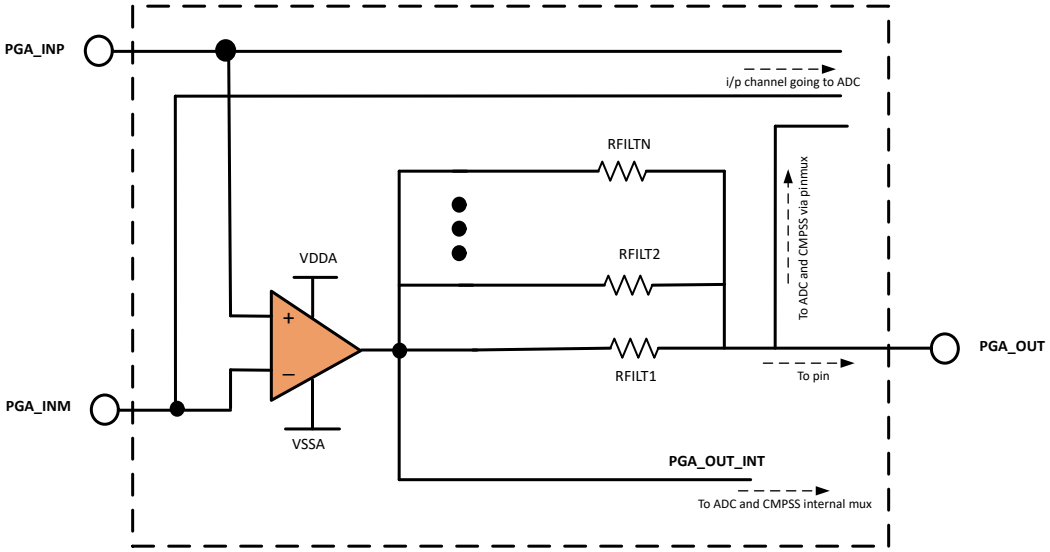


Figure 14-3. Standalone Mode

14.4.3 Non-inverting Mode

In non-inverting mode, the input voltage signal is applied directly to the non-inverting pin PGA_INP and generates an amplified output signal that is proportional and in phase with the input signal. The block diagram of PGA in this mode is illustrated in Figure 14-4. Feedback control of the non-inverting operational amplifier is achieved by applying a small part of the output voltage signal back to the inverting input terminal by way of a voltage divider network R_{ia} – R_{ib}. The voltage gain in this mode can be calculated using Equation 11:

$$V_{PGA\_OUT} = \left(1 + \frac{R_{ib}}{R_{ia}}\right) V_{PGA\_INP} \quad (11)$$

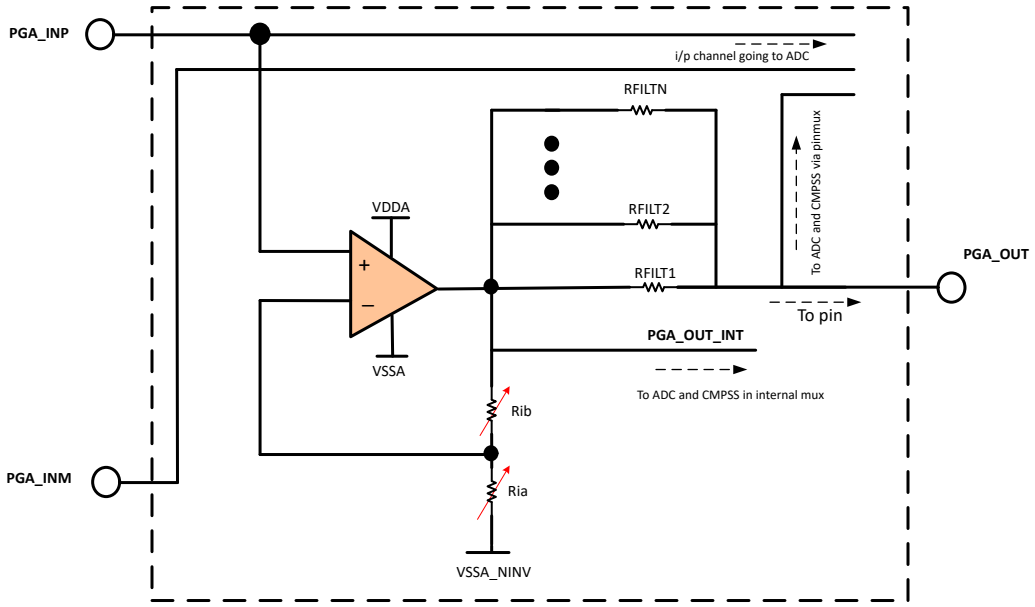


Figure 14-4. Non-inverting Mode

14.4.4 Subtractor Mode

In subtractor mode, the subtraction of the two input voltages is possible. Superposition is used to calculate the output voltage resulting from each input voltage, and then the two output voltages are subtracted to arrive at the final output voltage. The block diagram of PGA in this mode is demonstrated in Figure 14-5. The voltage gain can be derived as:

$$V_{PGA\_OUT} = \left(1 + \frac{R_{ib}}{R_{ia}}\right)V_{PGA\_INP} - \left(\frac{R_{ib}}{R_{ia}}\right)V_{PGA\_INM} \quad (12)$$

The differential signal is multiplied by the stage gain, so the subtractor mode name is a good choice for the circuit. If a voltage divider circuit with similar values of R_{ia}-R_{ib} is added to the non-inverting pin, this mode only amplifies the differential portion of the input signal; therefore, this mode rejects the common-mode portion of the input signal.

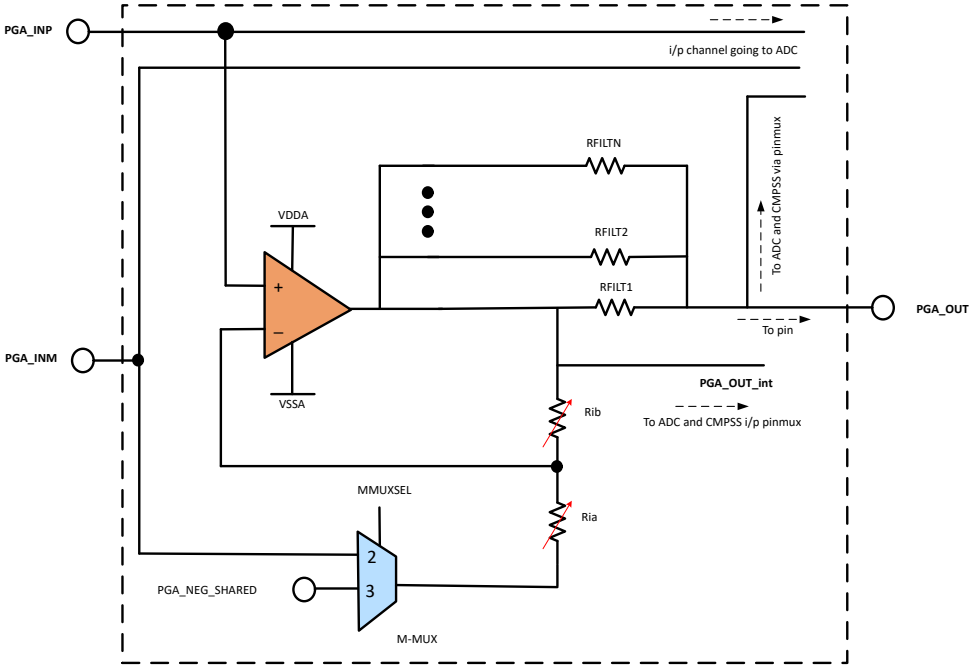


Figure 14-5. Subtractor Mode

14.5 External Filtering

To remove any unwanted high-frequency noise, two types of low-pass filter can be implemented using internal resistors of PGA module:

1. Low-pass filter using internal filter resistors R_{FILT} and external capacitor
2. Low-pass filter using internal gain resistors R_{ib} and external capacitor

14.5.1 Low-Pass Filter Using Internal Filter Resistor and External Capacitor

The PGA output can be routed to a pin through an embedded series resistor for the purpose of low-pass filtering the amplified signal. The filter resistance is software selectable using the PGACTL[FILT_RES_SEL] register field. The default selection of PGACTL[FILT_RES_SEL] = 0 disables the filter path.

The cutoff frequency can be estimated using the standard low-pass RC given by:

$$f_{cutoff} = \frac{1}{2\pi R_{FILT} C_{FILTER}} \quad (13)$$

Each gain mode requires a minimum amount of series resistance when filtering is enabled. The values are shown in Table 14-3. Also, the external capacitor value C_{FILTER} influences the ADC sampling performance. See Section 14.10.1.2 for more information.

Table 14-3. Minimum Filter Resistance

PGACTL[GAIN]	Minimum R_{FILT} Required PGACTL[FILT_RES_SEL]
0	50Ω
1	50Ω
2	50Ω
3	100Ω
4	100Ω
5	200Ω
6	400Ω

14.5.2 Single Pole Low-Pass Filter Using Internal Gain Resistor and External Capacitor

Some applications cannot tolerate having an RC network at the output of the amplifier. Amplifier output current flowing through the filter resistor creates a voltage offset that introduces output error. In this case, one can opt to filter the noise spikes by placing a feedback capacitor across the feedback loop. In fact, a simple single-pole low pass filter can be implemented by placing an external capacitance between PGA_INM and PGA_OUT in internal gain mode, which is shown in Figure 14-6.

This form of very simple filter is normally used in instances where a small amount of roll off is required. PGA feedback node RTAP can be brought on pin PGA_INM for external filtering by selecting PMUXSEL = 00, NMUXSEL = 01, and MMUXSEL = 01. Also, to enable this mode, PGA_FB_ON_PIN input/register needs to be set to 1.

The cutoff frequency for this type of filter can be calculated very easily by working out the frequency at which the reactance of the capacitor equals the resistance of the resistor R_{ib}, which is defined by Equation 14:

$$f_{cutoff} = \frac{1}{2\pi R_{ib} C_{Ext}} \tag{14}$$

The gain for this op amp circuit can be calculated in the normal way ignoring the effect of the capacitor:

$$Gain = \left(1 + \frac{R_{ib}}{R_{ia}}\right) \tag{15}$$

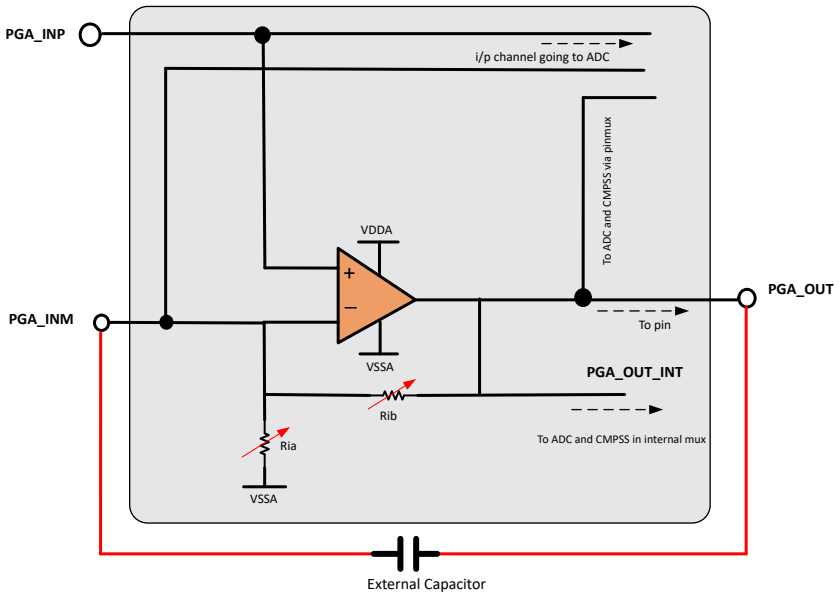


Figure 14-6. Low-Pass Filter Using External Capacitor

14.6 Error Calibration

Op-Amp error calibration is the process of adjusting the Op-Amp parameters to minimize or eliminate errors. Calibration can be done using different methods, depending on the type of error and the Op-Amp configuration. To reduce inherent offset, factory-generated values are written to the trim registers by calling the Device_cal() function that is located in TI reserved OTP.

14.6.1 Offset Error

This error occurs due to mismatches in the Op-Amp input terminals. The offset error appears as a constant DC offset across the PGA output range, see Figure 14-7. The built-in Device_cal() function reduces the offset error so that the error falls within the data sheet specifications.

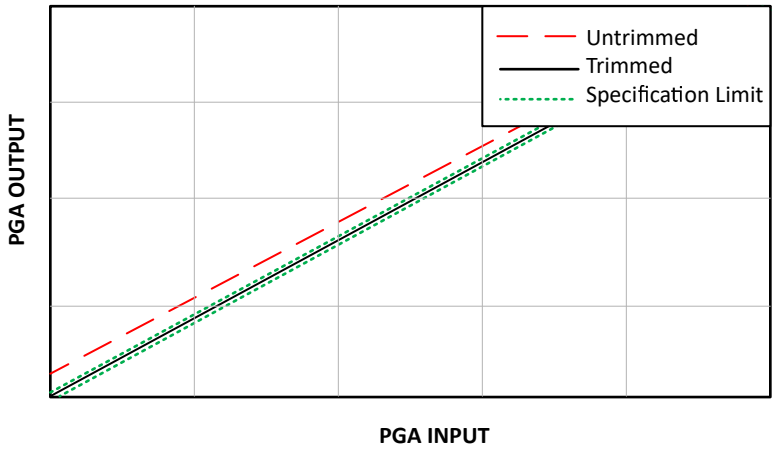


Figure 14-7. PGA Offset Trim

14.6.2 Gain Error

This error occurs due to mismatches in the Op-Amp gain. After the offset error has been removed, the remaining error, the gain error appears as a scaled error that increases in magnitude with increasing PGA output voltage, see Figure 14-8. The hardware trim targets the gain performance so that the hardware trim falls within the data sheet specifications.

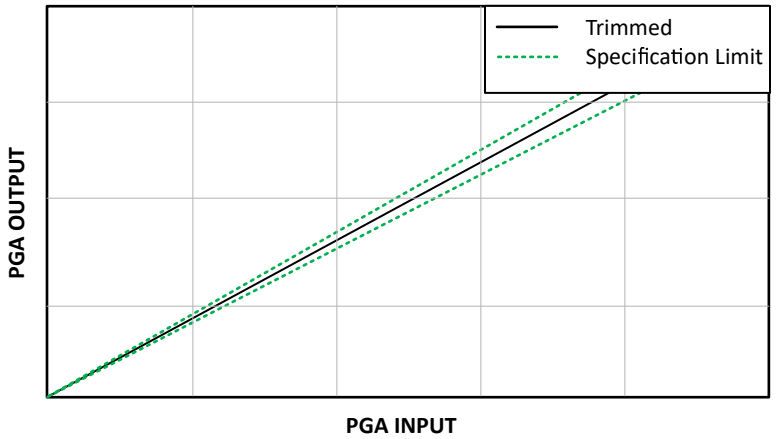


Figure 14-8. PGA Gain Error

14.7 Chopping Feature

The PGA supports chopping feature, which is a continuous-time modulation technique in which the signal and offset are modulated to different frequencies. The motivation behind adding this feature is primarily to reduce input referred offset to a very small number (in micro volt order). Not only chopping is dynamic technique that continuously reduces offset, but also this technique removes low frequency 1/f noise as well as offset drift over temperature or time.

The chopping feature is demonstrated in Figure 14-9. In broad terms, an input voltage first passes through a chopper driven by a clock at frequency f_{ch}; so, the voltage is converted to a square-wave voltage. Then, the modulated signal along with the offset are amplified by the fully differential amplifier. The second chopper then demodulates the amplified input signal back to DC, while the offset is shifted to the frequency of the clock f_{ch} harmonics. The DC offset can be filtered out by a low-pass filter.

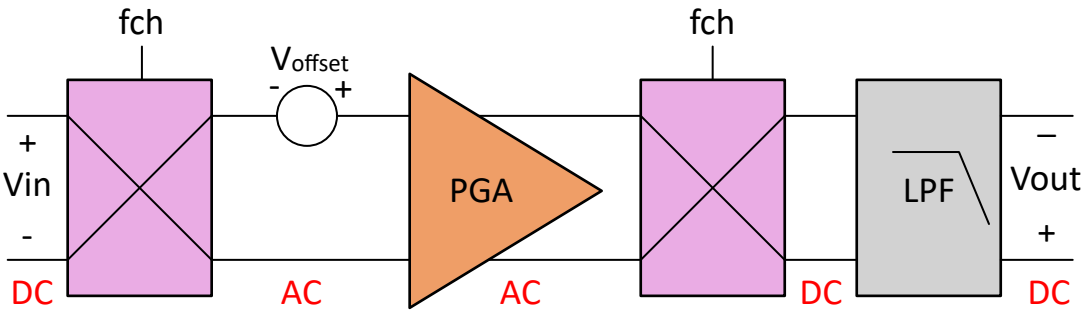


Figure 14-9. General Chopping Technique

This device supports ADC-assisted method. In this method, the chopping control signal comes from the on-chip ADC module. Moreover, Start of Conversion (SOC) in the ADC module can be used to implement chopping. In this method, PGA output is sampled by the on-chip ADC module. Note that the ADC-assisted method relies on ADC module for filtering the output instead of using external low-pass filter. To remove the offset, an average of the output signal can be taken in ADC module. This functionality is provided in post-processing block in ADC module. (See Chapter 12). The ADC-assisted chopping stands out from purely analog chopping methods due to the ability to offer higher accuracy and precision.

Figure 14-10 shows the general diagram of ADC-assisted chopping mode, in which a logic block is used to generate the necessary signal to handle chopping. To configure the ADC-assisted chopping, write to the PGA_CHOP field in the MUXSEL register. To configure control signal for ADC-assisted chopping, write to the CHOP_EXTCTRL field in the PGACTL register.

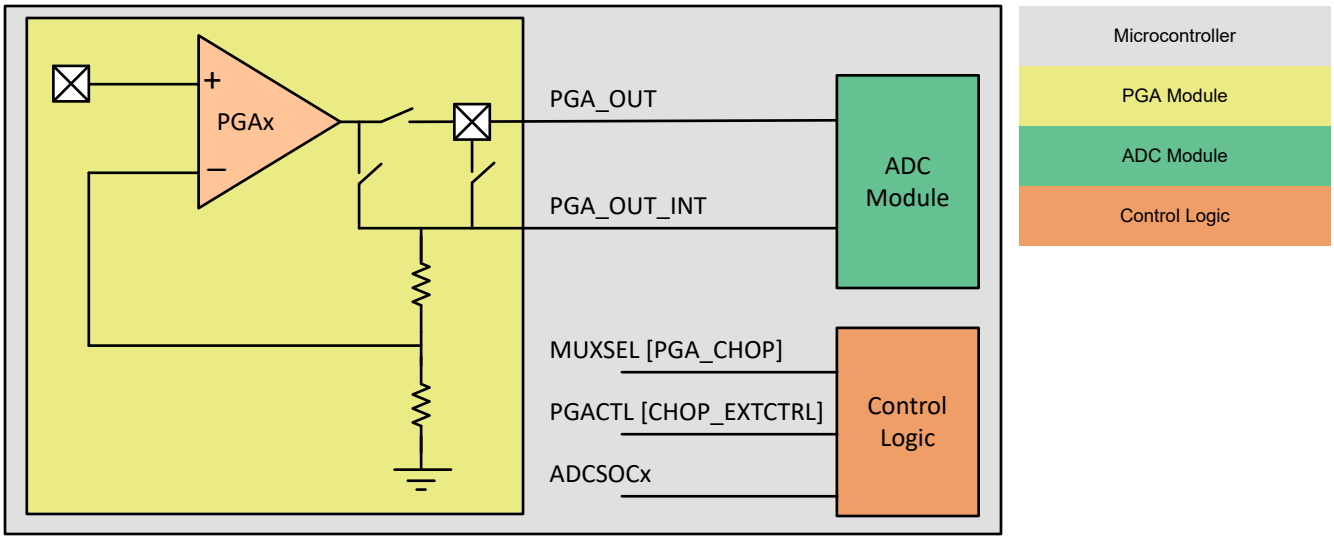


Figure 14-10. ADC-Assisted Chopping Block Diagram

For the PGA, the ADC assisted chopping feature is supported when the ADC is connected to the PGA_OUT_INT of the PGA. Table 14-4 shows the connectivity between the PGA and ADC modules.

Table 14-4. PGA and ADC Connection

PGA Connectivity to ADC	ADC
PGA1_OUT_INT	A25

14.8 Enabling and Disabling the PGA Clock

If the clock to the PGA is disabled while the PGA is outputting a voltage, the output voltage remains unaffected but the PGA registers are no longer updated with register writes. Enabling the clock resumes register writes.

14.9 Lock Register

The PGALOCK register provides the ability to block writes to certain PGA configuration registers. Locking register writes can prevent spurious or malicious code from modifying critical register settings. Once a register has been locked using the PGALOCK register, only a module-level or device-level reset can restore write functionality.

14.10 Analog Front-End Integration

The PGA operates together with the other embedded analog modules (ADC, CMPSS) creating an analog front-end system.

14.10.1 Analog-to-Digital Converter (ADC)

In the simplest application, the PGA amplifies small input signals to increase the ADC dynamic range. The PGA also provides the additional benefit of buffering input signals from the ADC sample and hold capacitor, which further reduces sampling error.

Both the filtered and unfiltered PGA output paths are available for ADC sampling. Minimum ADC acquisition windows are recommended when sampling the PGA paths.

14.10.1.1 Unfiltered Acquisition Window

The device data sheet provides a minimum estimated ADC acquisition window for sampling the PGA_OUT signal with one ADC. This estimated value can provide sampling accuracy close to the specified performance parameters of the ADC. A longer acquisition window can be used for better performance.

14.10.1.2 Filtered Acquisition Window

The minimum ADC acquisition window for sampling the PGA_OF filtered signal with one ADC varies based on the values of R_{FILTER} and C_{FILTER}. To make sure of good performance, choose a C_{FILTER} capacitor that is large enough to satisfy most of the ADC sample and hold capacitor (Ch) charge requirements. The C_{FILTER} value can be sized based on the acceptable amount of ADC sampling error (LSB_{Err}):

$$C_{FILTER} = \frac{C_h \times 4096}{LSB_{Err}} \quad (16)$$

14.10.2 Comparator Subsystem (CMPSS)

The PGA output can be monitored by a CMPSS module for trips above or below a reference voltage. Up to two independent reference thresholds per CMPSS can be used for trip detection.

Both the filtered and unfiltered PGA output paths are available for CMPSS trip monitoring.

14.10.3 Alternate Functions

The PGA has up to three device terminals for operation: non-inverting pin PGA_INP, inverting pin PGA_INM, and output pin PGA_OUT, see Figure 14-11. Alternate paths at the device level allow the input and output filter terminals to take on alternate functions. This can be especially valuable, if the respective PGA resource is not used for an application. See Chapter 11 for more information.

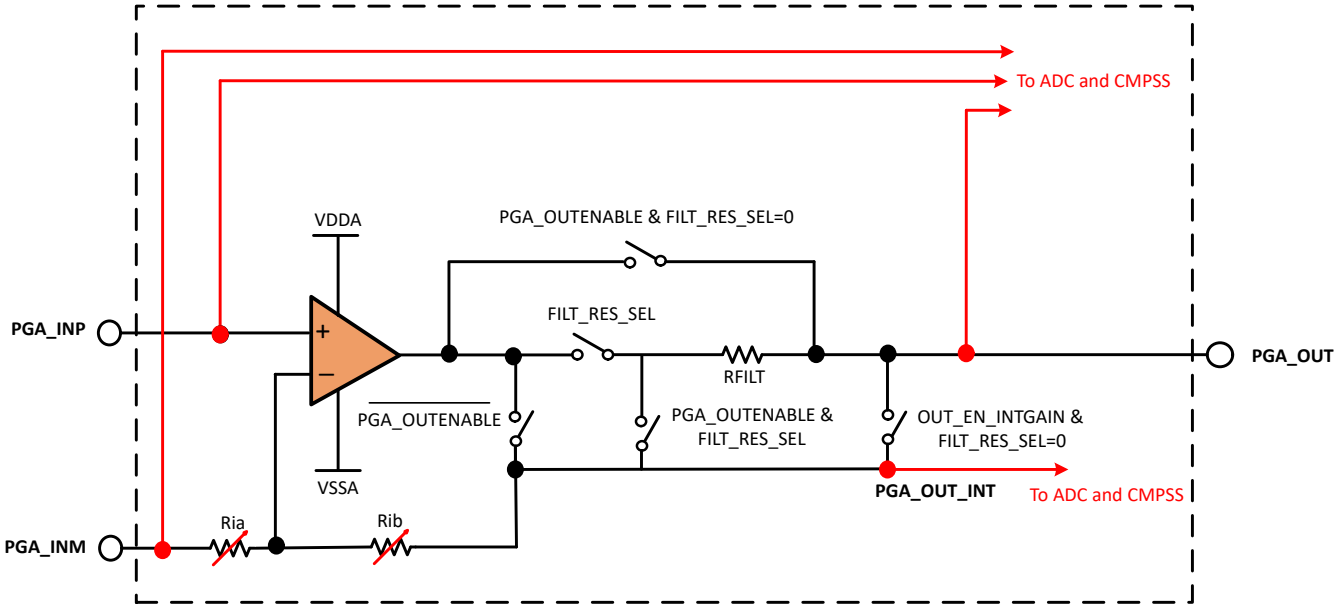


Figure 14-11. PGA Pin Alternate Functions

14.11 Examples

14.11.1 Non-Inverting Amplifier Using Non-Inverting Mode

Given a small positive signal, the PGA can be used to amplify the signal to increase the dynamic range of ADC sampling and comparator trip monitoring. For example, an input signal with a valid range between 0.25V to 0.75V can be amplified in 4× mode to produce an output signal between 1V to 3V.

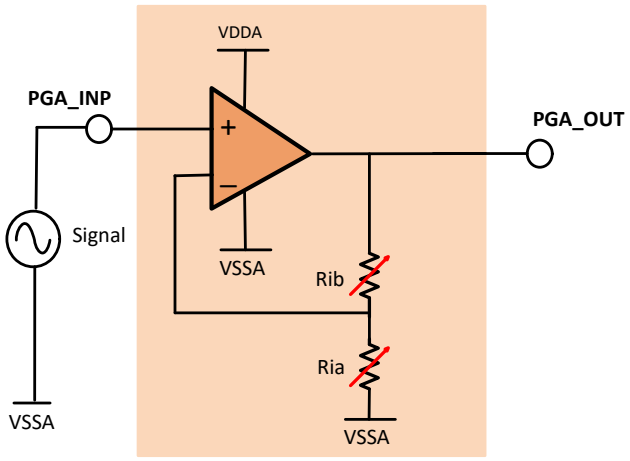


Figure 14-12. Signal Conditioning Using Non-inverting Mode

The amplified output voltage is calculated as:

$$V_{PGA\_OUT} = \left(1 + \frac{R_{ib}}{R_{ib}}\right) V_{Signal} \quad (17)$$

14.11.2 Buffer Mode

Assume that there is an analog sensor that gives a 0 to 12V range signal. To sample the signal using the ADC, a voltage divider along with a unity-gain op-amp is required. In the circuit shown in Figure 14-13, the voltage divider (36kΩ and 100kΩ) brings the 12V sensor voltage down to something less than 3.3V. The 100Ω filter resistor and 1nF capacitor form a low-pass filter with a cut-off frequency of 1.5MHz.

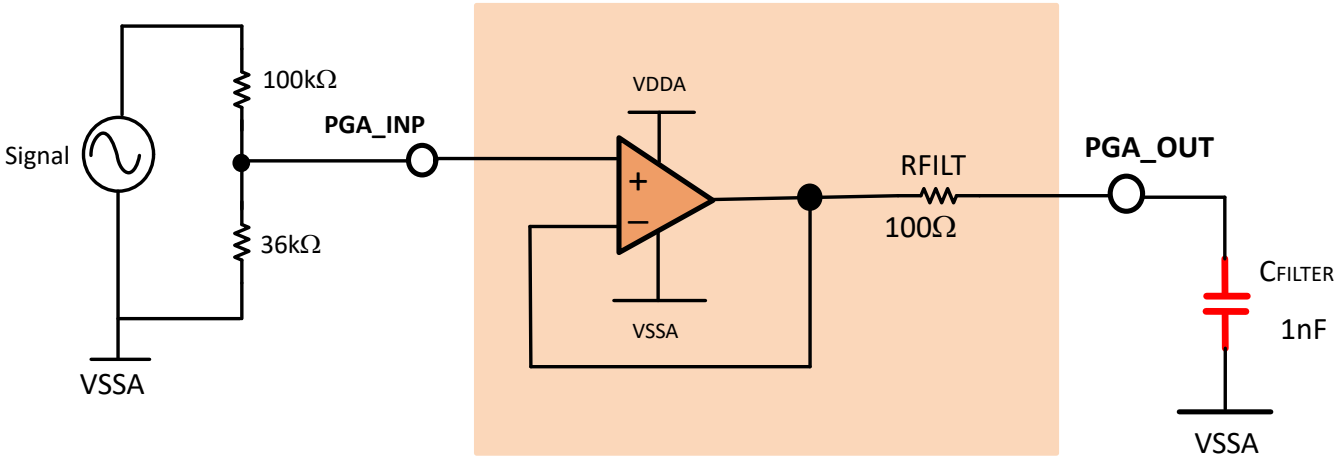


Figure 14-13. Buffer Mode Example

14.11.3 Low-Side Current Sensing

Low-side current sensing connects the sensing resistor between the load and ground. To not adversely affect current flow, the current resistors have a small value that produces a proportionally small voltage across them. The small voltage across the shunt resistor must usually be boosted from tens or hundreds of millivolts to tenths of a volt or volts for upstream conversion by an analog-to-digital converter (ADC). This task can be performed by PGA. Figure 14-14 shows the low-side current sensing using PGA.

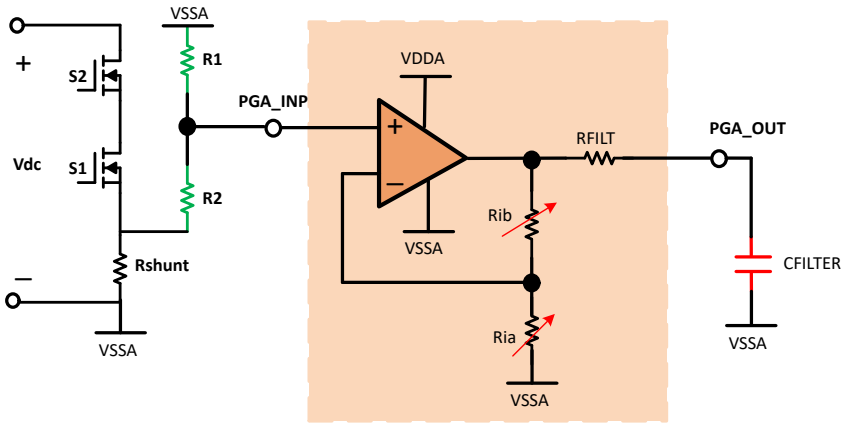


Figure 14-14. PGA Shunt Current Example

In this circuit, the amplifier gain is set by the ratio of R_{ib} divided by R_{ia}. If the divider resistors R_1=R_{ib} and R_2=R_{ia}, then the amplified voltage can be calculated by:

$$V_{PGA\_OUT} = \frac{R_{ib}}{R_{ia}}\left(R_{shunt}I_{load}\right) \quad (18)$$

If low-pass filtering is desired, an external capacitor in conjunction with internal filter resistors R_{FILT} can be used. The filter cutoff frequency is estimated using:

$$f_{cutoff} = \frac{1}{2\pi R_{FILT}C_{FILTER}} \quad (19)$$

14.11.4 Bidirectional Current Sensing

The PGA senses current flow through a sense resistor in both directions, demonstrated in Figure 14-15. The bidirectional current-sensing capability is achieved by applying a positive voltage at the non-inverting pin to offset the output voltage. A positive differential voltage sensed at the inputs results in an output voltage that is greater than the applied reference voltage; likewise, a negative differential voltage at the inputs results in output voltage that is less than the applied reference voltage. The output voltage of the PGA is given by:

$$V_{PGA\_OUT} = \frac{R_{ib}}{R_{ia}}(R_{sense}I_{load}) + V_{REF} \quad (20)$$

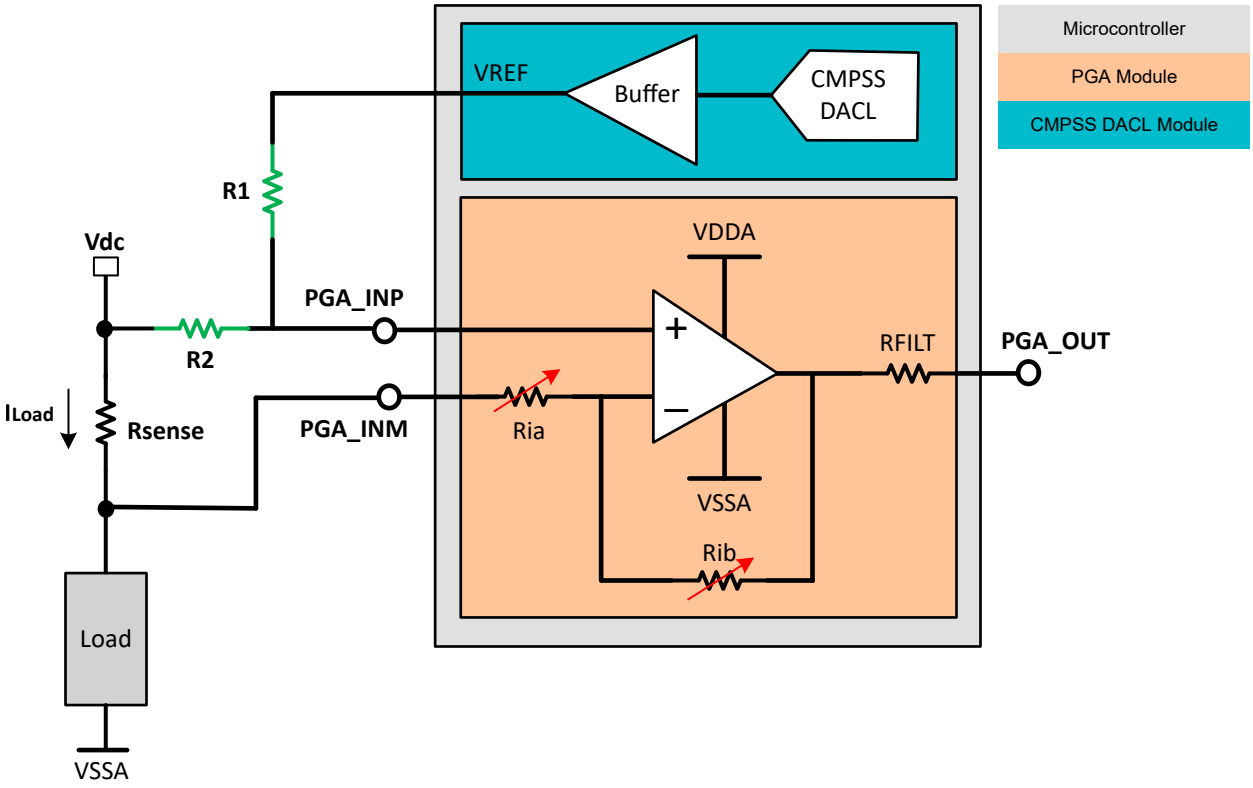


Figure 14-15. Bidirectional Current Sensing

14.12 Software

14.12.1 PGA Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/pga

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

14.12.1.1 PGA CMPSSDAC-ADC External Loopback Example

FILE: pga_ex1_dac_adc_ext_loopback.c

This example generates 800 mV using the CMPSSDAC output. CMPSSLITE is used route the low DAC output to an external pin to be used as an external DAC. The output of the CMPSS LITE DACL is externally connected to PGA for 2x gain amplification. It uses two ADC channels to sample the CMPSSDAC output and the amplified voltage output from PGA. The ADC is connected internally to these signals.

External Connections

- Connect CMPSSLITE_DACL_OUT (Analog Pin A0) to PGA1_INP1 (Analog Pin A11)
- Connect PGA1_INM1 (Analog Pin A4) to GND

Watch Variables

- *dacResult* - The DAC output voltage.
- *pgaResult* - The amplified DAC voltage.
- *pgaGain* - The ratio of the amplified DAC voltage to the original DAC output. This should always read a value of ~2.0.

14.13 PGA Registers

This Section describes the PGA Registers.

14.13.1 PGA Base Address Table

Table 14-5. PGA Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
Pga1Regs	PGA_REGS	PGA1_BASE	0x0000_5B00	YES	YES

14.13.2 PGA_REGS Registers

Table 14-6 lists the memory-mapped registers for the PGA_REGS registers. All register offset addresses not listed in Table 14-6 should be considered as reserved locations and the register contents should not be modified.

Table 14-6. PGA_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	PGACTL	PGA Control Register	EALLOW,LOCK: PGALOCK_TYPE2.PGACTL
1h	MUXSEL	Mux Selection Register	EALLOW,LOCK: PGALOCK_TYPE2.MUXSEL
2h	OFFSETTRIM	Offset Trim Register	EALLOW,LOCK: PGALOCK_TYPE2.OFFSETTRIM
5h	PGATYPE	PGA Type Register	
6h	PGALOCK	PGA Lock Register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 14-7 shows the codes that are used for access types in this section.

Table 14-7. PGA_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

14.13.2.1 PGACTL Register (Offset = 0h) [Reset = 0000h]

PGACTL is shown in Figure 14-16 and described in Table 14-8.

Return to the Summary Table.

PGA Control Register

Figure 14-16. PGACTL Register

15	14	13	12	11	10	9	8
CHOP_EXTCTRL	RESERVED					PGA_OUTEN_INTGAIN	PGA_OUTENABLE
R/W-0h	R-0h					R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GAIN			RESERVED	FILT_RES_SEL			PGAEN
R/W-0h			R-0h	R/W-0h			R/W-0h

Table 14-8. PGACTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CHOP_EXTCTRL	R/W	0h	CHOP Signal Control 0 Chop signal is low 1 Chop signal is high Reset type: SYSRSn
14-10	RESERVED	R	0h	Reserved
9	PGA_OUTEN_INTGAIN	R/W	0h	PGA Internal Gain on Pin 0 PGA internal gain path comes to a pin 1 PGA nternal gain path does not come to a pin Reset type: SYSRSn
8	PGA_OUTENABLE	R/W	0h	PGA Output Enable 0 PGA Output does not come to a pin 1 PGA Output does come to a pin Reset type: SYSRSn
7-5	GAIN	R/W	0h	PGA Gain programming-1-64 000 x 1 001 x 2/-1 010 x 4/-3 011 x 8/-7 100 x 16/-15 101 x 32/-31 110 x 64/-63 111 reserved Reset type: SYSRSn
4	RESERVED	R	0h	Reserved
3-1	FILT_RES_SEL	R/W	0h	Filter Resistor Select 000 Filter Disabled 001 50 010 100 011 200 100 400 101 800 110 Filter Disabled 111 Filter Disabled Reset type: SYSRSn
0	PGAEN	R/W	0h	PGA Enable. 0 PGA is disabled and powered down. 1 PGA is enabled. Reset type: SYSRSn

14.13.2.2 MUXSEL Register (Offset = 1h) [Reset = 0000h]

MUXSEL is shown in Figure 14-17 and described in Table 14-9.

Return to the Summary Table.

Mux Selection Register

Figure 14-17. MUXSEL Register

15	14	13	12	11	10	9	8
RESERVED	PGA_CHOP		PGA_FBONPIN	RESERVED	MMUXSEL		
R-0h	R/W-0h		R/W-0h	R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	NMUXSEL			RESERVED		PMUXSEL	
R-0h	R/W-0h			R-0h		R/W-0h	

Table 14-9. MUXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14-13	PGA_CHOP	R/W	0h	PGA Output Chopping Control 00 Chopping Disabled 01 Reserved 10 ADC assisted chop (using ADCSOC as ctrl) 11 ADC assisted chop (using PGA_CHOP_EXTCTRL register0 Reset type: SYSRSn
12	PGA_FBONPIN	R/W	0h	PGA Feedback to Negative Input Connection 0 PGA_INM0 and Inverting Input are not connected 1 PGA_INM0 and Inverting Input are connected Reset type: SYSRSn
11	RESERVED	R	0h	Reserved
10-8	MMUXSEL	R/W	0h	PGA M Mux Select 0x0 Stand alone op-amp or G=1 mode select 0x1 Non-Inverting gain mode select, feeback resistors have path to VSSA 0x2 Inverting gain mode select, feedback resistors have path to local PGA_INM1 pin 0x3 Inverting gain mode select, feedback resistors have path to local PGA_INM2 pin 0x4 Reserved 0x5 Reserved 0x6 Reserved 0x7 Reserved Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-4	NMUXSEL	R/W	0h	PGA Negative Input Mux Select 0x0 Select RTOP (Inter Gain Resistor Tap Point) as inverting input 0x1 Select RTAP (Pre Gain Resistor Tap Point) as inverting input 0x2 Select PGA_INM1 pin as inverting input 0x3 Select PGA_INM2 pin as inverting input 0x4 Reserved 0x5 Reserved 0x6 Reserved 0x7 Reserved Reset type: SYSRSn
3-2	RESERVED	R	0h	Reserved

Table 14-9. MUXSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	PMUXSEL	R/W	0h	PGA Positive Input Mux Select 0x0 Select PGA_INP1 pin as non inverting input 0x1 Select PGA_INP2 pin as non inverting input 0x2 Select PGA_INP3 pin as non inverting input 0x3 Reserved Reset type: SYSRSn

14.13.2.3 OFFSETTRIM Register (Offset = 2h) [Reset = 00000000h]

OFFSETTRIM is shown in Figure 14-18 and described in Table 14-10.

Return to the Summary Table.

Offset Trim Register

Figure 14-18. OFFSETTRIM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								PGA_OFFSETTRIMP							
R-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PGA_OFFSETTRIMN							
R-0h								R/W-0h							

Table 14-10. OFFSETTRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	PGA_OFFSETTRIMP	R/W	0h	PGA internal (analog) offset trim feature for i/p PMOS pair Reset type: SYSRSn
15-8	RESERVED	R	0h	Reserved
7-0	PGA_OFFSETTRIMN	R/W	0h	PGA internal (analog) offset trim feature for i/p NMOS pair Reset type: SYSRSn

14.13.2.4 PGATYPE Register (Offset = 5h) [Reset = 0201h]

PGATYPE is shown in Figure 14-19 and described in Table 14-11.

Return to the Summary Table.

PGA Type Register

Figure 14-19. PGATYPE Register

15	14	13	12	11	10	9	8
TYPE							
R-2h							
7	6	5	4	3	2	1	0
REV							
R-1h							

Table 14-11. PGATYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	TYPE	R	2h	PGA Type. Reset type: SYSRSn
7-0	REV	R	1h	PGA Revision. Reset type: SYSRSn

14.13.2.5 PGALOCK Register (Offset = 6h) [Reset = 0000h]

PGALOCK is shown in Figure 14-20 and described in Table 14-12.

Return to the Summary Table.

PGA Lock Register

Figure 14-20. PGALOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	PGATMCTL	OFFSETTRIM	MUXSEL	PGACTL
WSonce-0h	R-0h	R-0h	R-0h	WSonce-0h	WSonce-0h	WSonce-0h	WSonce-0h

Table 14-12. PGALOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	RESERVED	WSonce	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	PGATMCTL	WSonce	0h	0 Writes to PGATMCTL are enabled. 1 Writes to PGATMCTL are disabled. Reset type: SYSRSn
2	OFFSETTRIM	WSonce	0h	0 Writes to OFFSETTRIM are enabled. 1 Writes to OFFSETTRIM are disabled. Reset type: SYSRSn
1	MUXSEL	WSonce	0h	0 Writes to MUXSEL are enabled. 1 Writes to MUXSEL are disabled. Reset type: SYSRSn
0	PGACTL	WSonce	0h	0 Writes to PGACTL are enabled. 1 Writes to PGACTL are disabled. Reset type: SYSRSn

Chapter 15

Multi-Channel Pulse Width Modulator (MCPWM)

The Multi-channel pulse-width modulator (MCPWM) peripheral is a key element in controlling many of the power electronic systems found in both commercial and industrial equipment. These systems include digital motor control, switch mode power supply control, uninterruptible power supplies (UPS), and other forms of power conversion. The MCPWM peripheral can also perform a digital-to-analog (DAC) function, where the duty cycle is equivalent to a DAC analog value; sometimes referred to as a power DAC.

15.1 Introduction

This chapter includes an overview and information about each submodule:

- Time Base (TB) Submodule
- Counter Compare (CC) Submodule
- Action Qualifier (AQ) Submodule
- Dead-Band Generator (DB) Submodule
- Trip Zone (TZ) Submodule
- Event Trigger (ET) Submodule

The MCPWM module is a reduced version of the type-4 EPWM module featured on other C2000 devices. MCPWM has the following enhancements/removals compared to type 4 EPWM:

- **Memory Mapped Active/Shadow Registers:** Additional registers have been added to view the contents of the active register and the shadow register separately for CMPx, TBPRD, DBRED, DBFED, AQCTLA, and AQCTLB.
- **6 Channels per MCPWM module:** In contrast to EPWM, the MCPWM module can feature up to 6 channels on a single module. The 6 channels are treated as 3 pairs of signals, similar to 3 EPWM modules; however, some settings are shared across all 3 channel pairs such as TBPRD, DBRED, DBFED, and TBPHS. This can reduce design flexibility compared to 3 separate EPWM modules.
- **Removed Submodules/feature:** When compared to type-4 EPWM, the following features/submodules have been removed on MCPWM:
  - HRPWM
  - Separate interrupts for TZ events
  - Down-Count mode
  - Digital Compare submodule
  - Chopper module
  - EPWMXLINK
  - T1/T2 action qualifier events
  - Dead-band half-cycle clocking mode

Refer to the *EPWM to MCPWM Migration Guide* for more detail on feature changes and subsequent workarounds when migrating from EPWM to MCPWM.

The MCPWM peripheral is capable of generating complex pulse width waveforms with minimal CPU overhead or intervention. Similar to type-4 EPWM, the MCPWM is split into separate modules each with an independent function. This chapter is divided into individual sections by each MCPWM submodule. For most PWM applications, all modules in MCPWM must be understood and utilized to generate the desired PWM output.

In this document, the letters x and y within a signal or submodule name is used to indicate a generic MCPWM instance and channel pair on a device. For example, output signals MCPWMx_yA and MCPWMx_yB refer to the output signals from the MCPWMx instance and y channel pair (note that there are up to 3 channel pairs per MCPWM instance). Thus, MCPWM1_1A and MCPWM1_1B belong to MCPWM1 channel pair 1 and likewise MCPWM1_3A and MCPWM1_3B belong to MCPWM1 channel pair 3.

15.1.1 PWM Related Collateral

Foundational Materials

- *Real-Time Control Reference Guide*
  - Refer to the PWM section

Getting Started Materials

- *C2000 ePWM Developer's Guide Application Report*
- *Flexible PWMs Enable Multi-Axis Drives, Multi-Level Inverters Application Report*
- Getting Started with the C2000 ePWM Module (Video)

- *Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x Digital Signal Control Application Report*
  - Chapters 1 to 6 are Fundamental material, derivations, and explanations that are useful for learning about how PWM can be used to implement a DAC. Subsequent chapters are Getting Started and Expert material for implementing in a system.
- *Using the Enhanced Pulse Width Modulator (ePWM) Module Application Report*

Expert Materials

- C2000 real-time microcontrollers - Reference designs
  - See TI designs related to specific end applications used.

15.1.2 Submodule Overview

The MCPWM module consists of up to 3 complete PWM channel pairs with the following signal names:

- MCPWMx_1A
- MCPWMx_1B
- MCPWMx_2A
- MCPWMx_2B
- MCPWMx_3A
- MCPWMx_3B

Multiple MCPWM modules can exist on a single device, with each module denoted by a different value of "x".

Each MCPWM instance is identical with one exception. Some instances only include a single channel pair. Because of this, there are two types of MCPWM, 6-channel MCPWM (6CH) and 2-channel MCPWM (2CH). The MCPWM instances and the types are as follows:

MCPWM Instance	Number of Channels
MCPWM1	6
MCPWM3	2

Note that not all channels need to be used in either 6-channel or 2-channel MCPWM. For example, to use only 4 channels from a 6-channel MCPWM instance, simply configure the GPIO mux to avoid routing the remaining two channels to any device pin.

The MCPWM modules are chained together by way of a clock synchronization scheme that allows them to operate as a single system when required. Additionally, this synchronization scheme can be extended to the capture peripheral submodules (eCAP).

Each MCPWM module supports the following features:

- Dedicated 16-bit time-base counter with period and frequency control
- Up to 6 PWM outputs(3 channel pairs). Each channel pair can be used in the following configurations:
  - Two independent PWM outputs with single-edge operation
  - Two independent PWM outputs with dual-edge symmetric operation
  - One independent PWM output with dual-edge asymmetric operation
- Asynchronous override control of PWM signals through software.
- Programmable phase-control support for lag or lead operation relative to other MCPWM modules.
- Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.
- A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.
- All events can trigger both CPU interrupts and ADC start of conversion (SOC)
- Programmable event prescaling minimizes CPU overhead on interrupts.

Each MCPWM module consists of 6 submodules and is connected within a system by way of the signals shown in Figure 15-1.

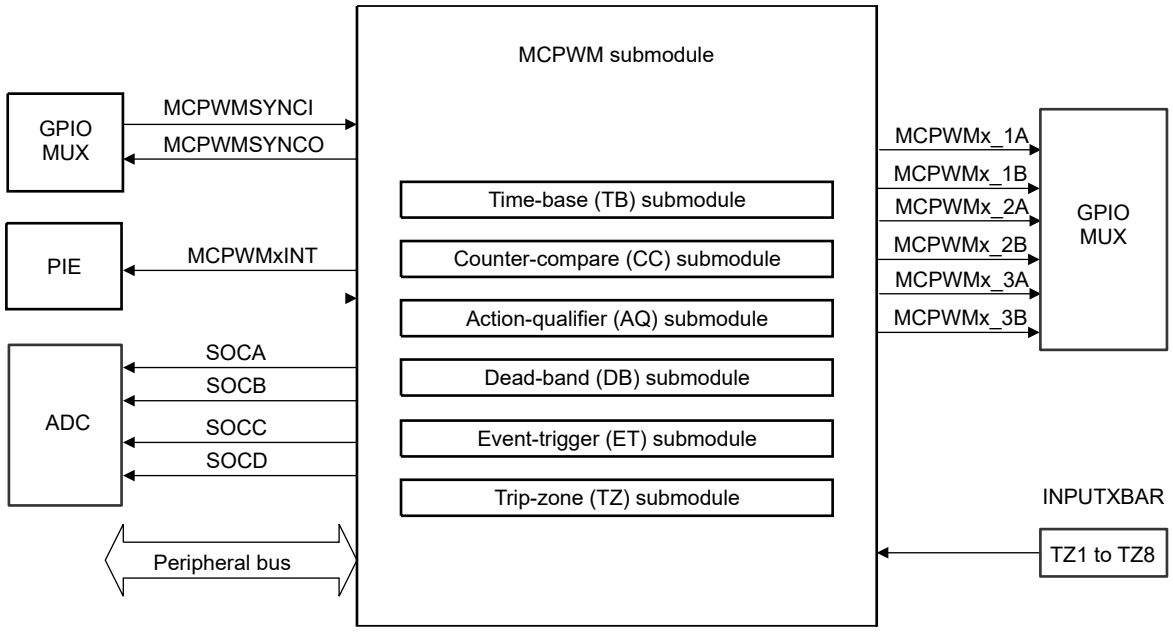


Figure 15-1. Submodules and Signal Connections for an MCPWM Module

Figure 15-2 shows more internal details of a single MCPWM module. The main signals used by the MCPWM module are:

- **MCPWM output signals (MCPWMx_yA and MCPWMx_yB):** The MCPWM output signals are made available external to the device.
- **Trip-zone signals ($\overline{TZ1}$ to $\overline{TZ8}$):** These input signals alert the MCPWM module of fault conditions external to the MCPWM module. Each submodule on a device can be configured to either use or ignore any of the trip-zone signals. Each TZx signal is directly connected to the output of one of the 8 MCPWM X-BAR outputs.
- **Time-base synchronization input (MCPWMxSYNCI), output (MCPWMxSYNCO), and peripheral (MCPWMxSYNCPER) signals:** For more information, see Section 15.4.3.3.

  Each MCPWM module also generates another PWMSYNC signal called MCPWMxSYNCPER. MCPWMxSYNCPER goes to the CMPSS for synchronization purposes. Functionality is configured using the TBCTL register. For more information on how MCPWMxSYNCPER is used by the CMPSS, see the *Comparator Subsystem (CMPSS)* chapter.
- **ADC start-of-conversion signals:** Each MCPWM module has four ADC start of conversion signals(SOCA, SOCB, SOCC, SOCD). Any MCPWM module can trigger a start of conversion. Whichever event triggers the start of conversion is configured in the event-trigger submodule of the MCPWM.
- **Comparator output signals (COMPxOUT):** Output signals from the comparator module can be fed through the MCPWM X-BAR to one or all of the 8 trip inputs to induce one-shot or cycle-by-cycle trip events. Refer to the *Crossbar (X-BAR)* chapter and the Trip-Zone submodule section of this chapter for more information.
- **Peripheral bus:** The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the MCPWM register file.

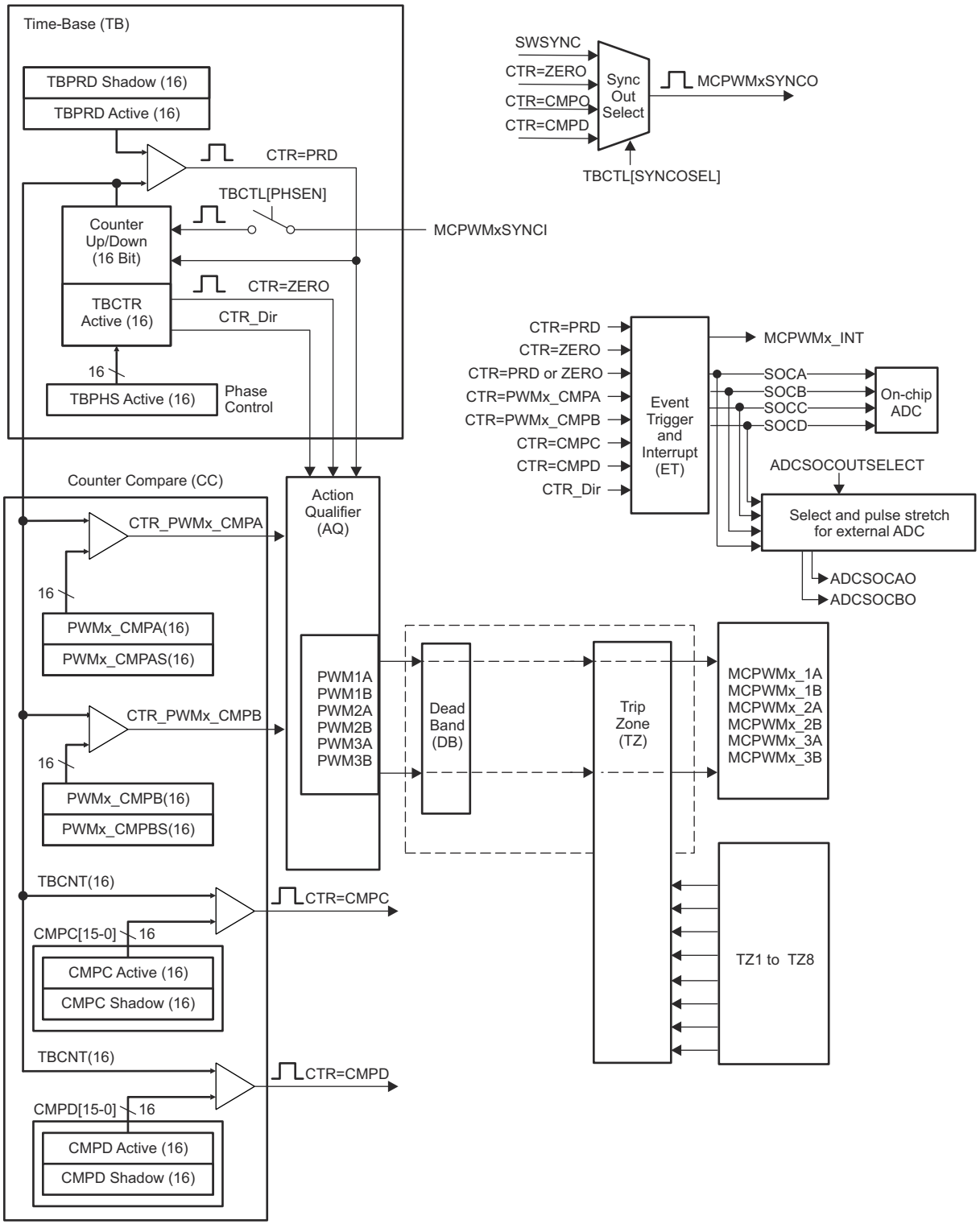


Figure 15-2. MCPWM Modules and Critical Internal Signal Interconnects

15.2 Configuring Device Pins

To connect the device input pins to the MCPWM module, the Input X-BAR and PWM X-BAR must be used. Some examples of when an external signal can be needed are TZx and MCPWMxSYNCI. Any GPIO on the device can be configured as an input. The GPIO input qualification can be set to asynchronous mode by setting the appropriate GPxQSEL register bits to 11. The internal pullups can be configured in the GPyPUD register. Since the GPIO mode is used, the GPyINV register can invert the signals. All trip signals must be routed through the PWM X-BAR, which can take in the Input X-BAR as an input to utilize an external trip signal.

The GPIO mux registers must be configured for this peripheral. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux, GPIO settings, and X-BAR configuration.

15.3 MCPWM Modules Overview

The 6 submodules are included in every MCPWM peripheral. Each of these submodules performs specific tasks that can be configured by software.

Table 15-1 lists the key submodules together with a list of the main configuration parameters. For example, if you need to adjust or control the duty cycle of a PWM waveform, see the counter-compare submodule in Section 15.5 for relevant details.

Table 15-1. Submodule Configuration Parameters

Submodule	Configuration Parameter or Option
Time Base (TB)	• Scale the time-base clock (TBCLK) relative to the MCPWM clock (MCPWMCLK). • Configure the PWM time-base counter (TBCTR) frequency or period. • Set the mode for the time-base counter: – count-up mode: used for asymmetric PWM – count-up-and-down mode: used for symmetric PWM • Configure the time-base phase relative to another MCPWM module. • Synchronize the time-base counter between modules through hardware or software. • Configure the direction (up or down) of the time-base counter after a synchronization event. • Configure how the time-base counter behaves when the device is halted by an emulator. • Specify the source for the synchronization output of the MCPWM module. • Configure one shot and global load of registers in this module.
Counter Compare (CC)	• Specify the PWM duty cycle for each outputs MCPWMx_yA and outputs MCPWMx_yB • Specify the time at which switching events occur on the MCPWMx_yA or MCPWMx_yB outputs. • Specify the programmable delay for interrupt and SOC generation with additional comparators • Configure one shot and global load of registers in this module.
Action Qualifier (AQ)	• Specify the type of action taken when a time-base counter-compare – No action taken – Output MCPWMx_yA and MCPWMx_yB switched high – Output MCPWMx_yA and MCPWMx_yB switched low – Output MCPWMx_yA and MCPWMx_yB toggled • Force the PWM output state through software control • Configure and control the PWM dead band through software • Configure one shot and global load of registers in this module.

Table 15-1. Submodule Configuration Parameters (continued)

Submodule	Configuration Parameter or Option
Dead-Band Generator (DB)	• Control of traditional complementary dead-band relationship between upper and lower switches • Specify the output rising-edge-delay value • Specify the output falling-edge delay value • Bypass the dead-band module entirely. In this case the PWM waveform is passed through without modification. • Allow MCPWMxB phase shifting with respect to the MCPWMxA output. • Configure one shot and global load of registers in this module.
Trip Zone (TZ)	• Configure the MCPWM module to react to one, all, or none of the trip-zone signals or digital compare events. • Specify the trip action taken when a fault occurs: – Force outputs high – Force outputs low – Force outputs to a high-impedance state – Configure outputs to ignore any trip condition. • Configure how often the MCPWM reacts to each trip-zone signal: – One-shot – Cycle-by-cycle • Bypass the trip-zone module entirely. • Programmable option for cycle-by-cycle trip clear
Event Trigger (ET)	• Enable the MCPWM events that trigger an interrupt. • Enable MCPWM events that trigger an ADC start-of-conversion event. • Specify the rate at which events cause triggers (every occurrence or every 2nd or up to 15th occurrence) • Poll, set, or clear event flags

15.4 Time-Base (TB) Submodule

Each MCPWM module has a time-base submodule that determines all of the event timing for the MCPWM module. Built-in synchronization logic allows the time-base of multiple MCPWM modules to work together as a single system.

Figure 15-3 illustrates the time-base submodule within the MCPWM.

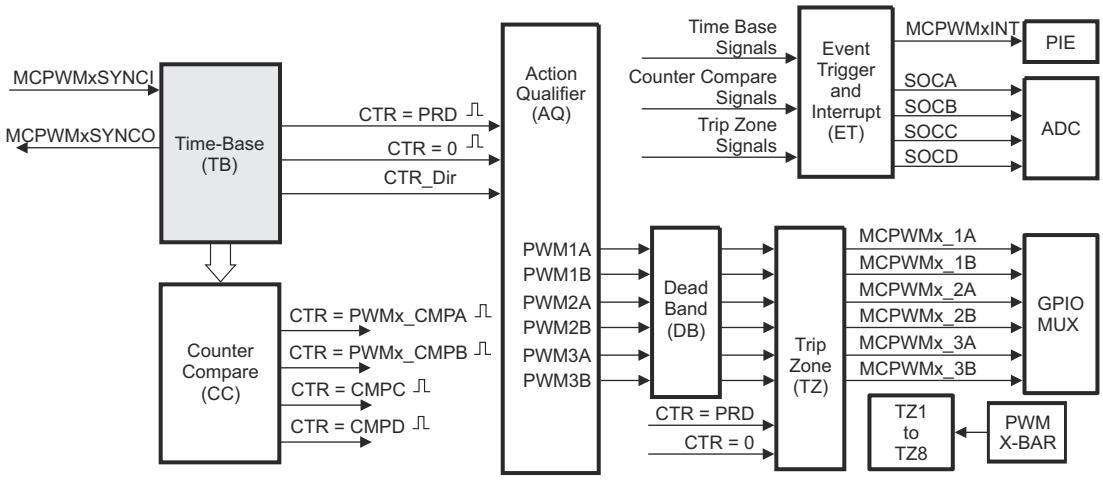


Figure 15-3. Time-Base Submodule

15.4.1 Purpose of the Time-Base Submodule

The time-base submodule can be configured for the following:

- Specify the MCPWM time-base counter (TBCTR) frequency or period to control how often events occur.
- Manage time-base synchronization with other MCPWM modules.
- Maintain a phase relationship with other MCPWM modules.
- Set the time-base counter to count-up, or count-up-and-down mode.
- Generate the following events:
  - CTR = PRD: Time-base counter equal to the specified period (TBCTR = TBPRD).
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00).
- Configure the rate of the time-base clock; a prescaled version of the MCPWM clock (MCPWMCLK). This allows the time-base counter to increment/decrement at a slower rate.

Note

If required by the application code to update the TBCTR value through software while the TBCTR is counting, note that the time-base module needs at least 1 TBCLK cycle for the time-base related events to be realized. Hence, the TBCTR can be written with TBCTR = PRD-1 instead of TBCTR = PRD (in case the counter is counting up) and can be written as TBCTR = 1 instead of TBCTR = 0 (in case the counter is counting down) for the events to be realized.

15.4.2 Controlling and Monitoring the Time-Base Submodule

The block diagram in Figure 15-4 shows the critical signals and registers of the time-base submodule. Table 15-2 provides descriptions of the key signals associated with the time-base submodule.

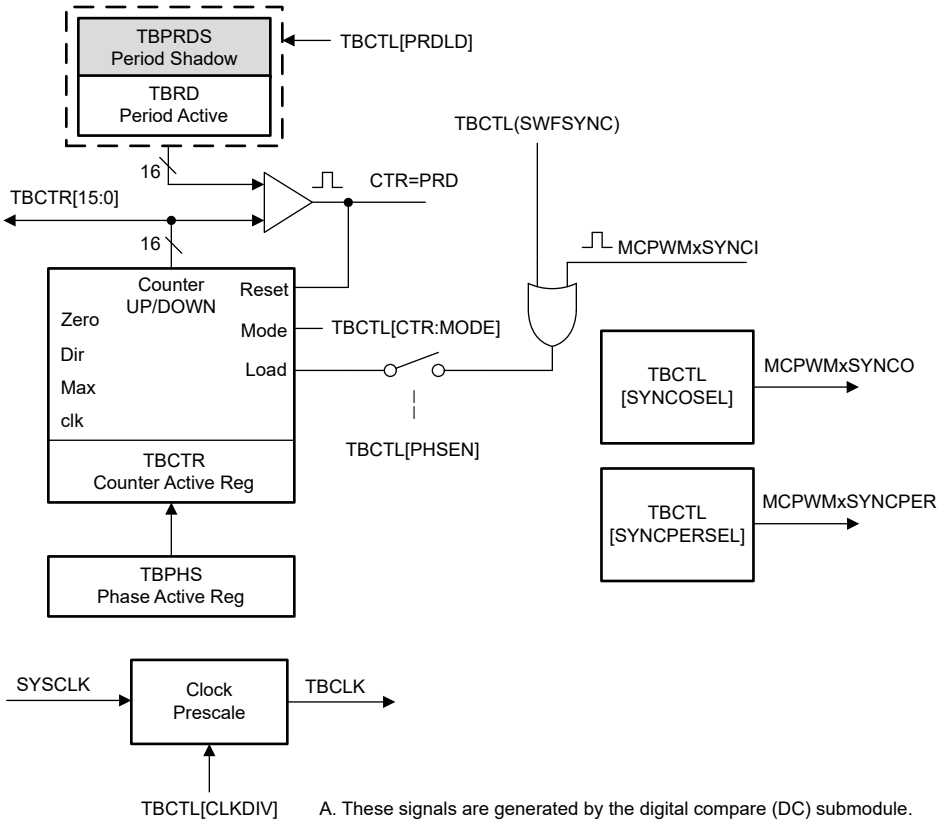


Figure 15-4. Time-Base Submodule Signals and Registers

Table 15-2. Key Time-Base Signals

Signal	Description
MCPWMxSYNCI	Time-base synchronization input. Input pulse used to synchronize the time-base counter with the counter of other MCPWM modules. For information on the synchronization order of a particular device, see Section 15.4.3.3.
MCPWMxSYNCO	Time-base synchronization output. This output pulse is used to synchronize the counter of other MCPWM modules. Using the TBCTL.SYNCOSEL bit, the source of the output pulse is selected.
MCPWMxSYNCPER	Time-base peripheral synchronization output. This output signal is used to synchronize the CMPSS to the MCPWM. The output signal can be configured using the TBCTL register.
CTR = PRD	Time-base counter equal to the specified period. This signal is generated whenever the counter value is equal to the active period register value. That is when TBCTR = TBPRD.
CTR = Zero	Time-base counter equal to zero. This signal is generated whenever the counter value is zero. That is when TBCTR equals 0x00.
CTR = PWMx_CMPB	Time-base counter equal to active counter-compare B register (TBCTR = PWMx_CMPB). This event is generated by the counter-compare submodule and used by the synchronization out logic. There is one CMPB register per PWM channel pair, denoted by "x".
CTRDIR	Time-base counter direction. Indicates the current direction of the MCPWM time-base counter. The signal is high when the counter is increasing and the signal is low when the counter is decreasing. The state of this signal can be observed in the TBSTS register.
TBCLK	Time-base clock. This is a prescaled version of the MCPWM clock (MCPWMCLK) and is used by all submodules within the MCPWM. This clock determines the rate at which time-base counter increments or decrements.

15.4.3 Calculating PWM Period and Frequency

The frequency of PWM events is controlled by the time-base period (TBPRD) register and the mode of the time-base counter. Figure 15-5 shows the period (T_{pwm}) and frequency (F_{pwm}) relationships for the up-count and up-down-count time-base counter modes when the period is set to 4 (TBPRD = 4). The time increment for each step is defined by the time-base clock (TBCLK) which is a prescaled version of the MCPWM clock (MCPWMCLK).

The time-base counter has two modes of operation selected by the time-base control register (TBCTL.CTRMODE):

- **Up-Down Count Mode:** In up-down count mode, the time-base counter starts from zero and increments until the period (TBPRD) value is reached. When the period value is reached, the time-base counter then decrements until the counter reaches zero. At this point, the counter repeats the pattern and begins to increment.
- **Up-Count Mode:** In up-count mode, the time-base counter starts from zero and increments until the counter reaches the value in the period register (TBPRD). When the period value is reached, the time-base counter resets to zero and begins to increment once again.

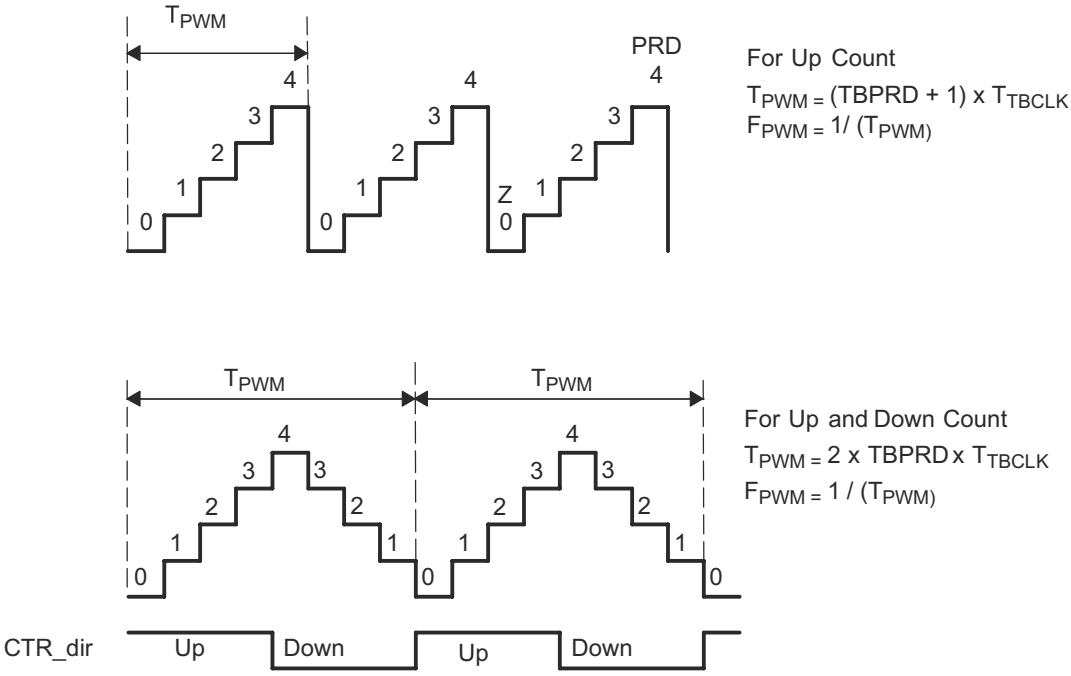


Figure 15-5. Time-Base Frequency and Period

15.4.3.1 Time-Base Period Shadow Register

The time-base period register (TBPRD) has a shadow register. Shadowing allows the register update to be synchronized with the hardware. The following definitions are used to describe all shadow registers in the MCPWM module:

- **Active Register:** The active register controls the hardware and is responsible for actions that the hardware causes or invokes.
- **Shadow Register:** The shadow register buffers provide a temporary holding location for the active register and have no direct effect on any control hardware. At a strategic point in time, the shadow register content is transferred to the active register. This prevents corruption or spurious operation due to the register being asynchronously modified by software.

In contrast to the EPWM module, MCPWM has two separate addresses for the active and shadow register. When shadow mode is enabled, write to the shadow register instead of the active register for shadow loading to occur. Writing to the active register immediately updates TBPRD. When shadowing is disabled, a write to the shadow register does not update the active register. Shadowing is enabled or disabled by the TBCTL[PRDLD] bit. This bit enables and disables the TBPRD shadow register as follows:

- **Time-Base Period Shadow Mode:** The TBPRD shadow register (TBPRDS) is enabled when TBCTL[PRDLD] = 0. When shadow mode is enabled, the value written to the shadow register is loaded to the active register on the next shadow load or global load event. The shadow register contents are transferred to the active register (TBPRD ← TBPRDS) when the time-base counter equals zero (TBCTR = 0x00). The PRDLDSYNC bit is valid only if TBCTL[PRDLD] = 0. A write to the active register (TBPRD) immediately updates TBPRD.

  The global load control mechanism can also be used with the time-base period register by configuring the appropriate bits in the global load configuration register (GLDCTL.GLD). When global load mode is selected the transfer of contents from shadow register to active register, for all registers that have a corresponding shadow register, occurs at the same event as defined by the configuration bits in Global Shadow to Active Load Control Register (GLDCTL.GLDMODE). Global load control mechanism is explained in Section 15.4.6.
- **Time-Base Period Immediate Load Mode:** If immediate load mode is selected (TBCTL[PRDLD] = 1), then a read from or a write to the TBPRD memory address goes directly to the active register.

15.4.3.2 Time-Base Clock Synchronization

The TBCLKSYNC bit in the peripheral clock enable registers allows all users to globally synchronize all enabled MCPWM modules to the time-base clock (TBCLK). When set, all enabled MCPWM module clocks are started with the first rising edge of TBCLK aligned. For synchronized TBCLKs, the prescalers for each MCPWM module must be set identically.

The proper procedure for enabling MCPWM clocks is as follows:

1. Set TBCLKSYNC = 0
2. Configure MCPWM modules
3. Set TBCLKSYNC = 1

15.4.3.3 Time-Base Counter Synchronization

The MCPWM synchronization scheme allows for increased flexibility of synchronization of the MCPWM modules. Each MCPWM module has a synchronization input (MCPWMxSYNCI), a synchronization output (MCPWMxSYNCO) and a peripheral synchronization output (MCPWMxSYNCPER).

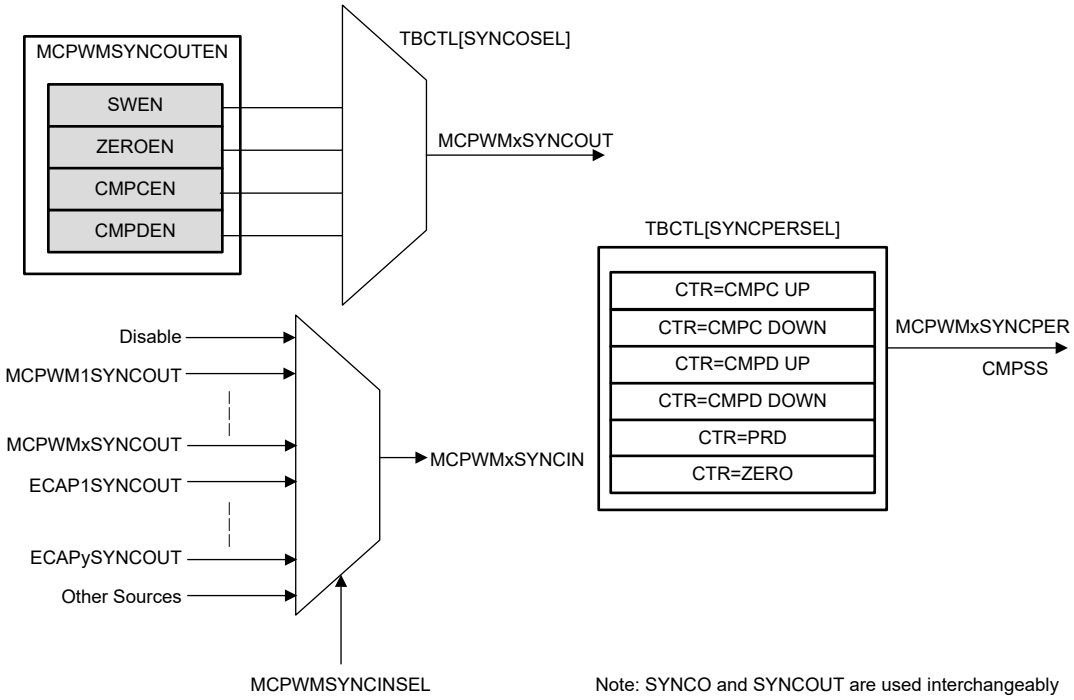
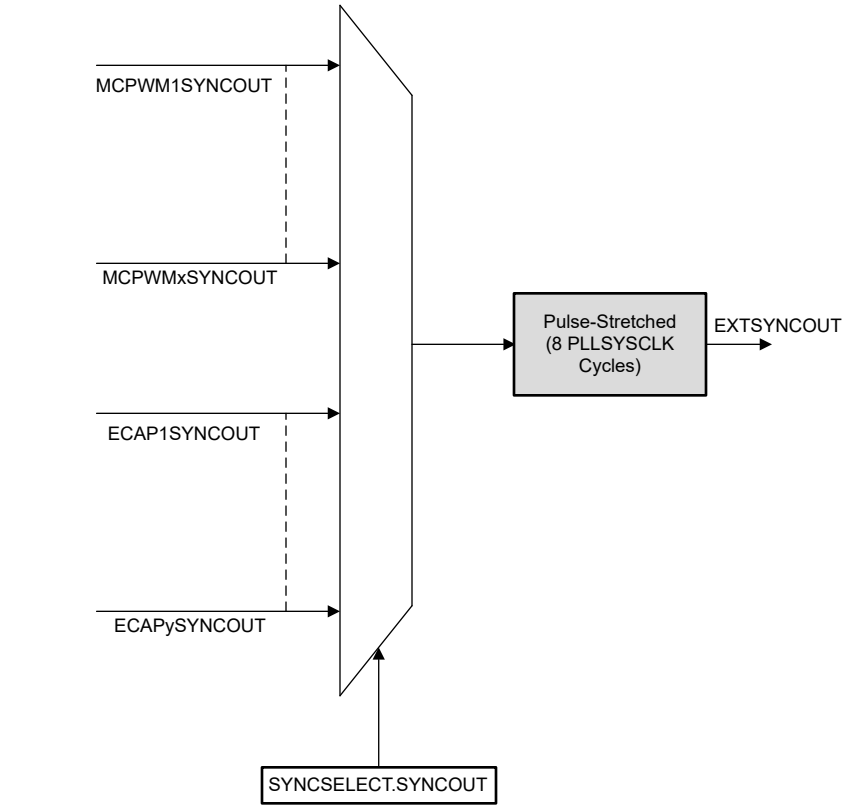


Figure 15-6. Time-Base Counter Synchronization Scheme



MCPWMxSYNCOUT (where x = number of MCPWM and y = number of ECAP modules)

Figure 15-7. MCPWM External SYNC Output

Note

See the data sheet for the number of MCPWM and eCAP modules available on your specific device.

Each MCPWM module can be configured to use or ignore the synchronization input. If the TBCTL[PHSEN] bit is set, then the time-base counter (TBCTR) of the MCPWM module is automatically loaded with the phase register (TBPHS) contents when one of the following conditions occur:

- **MCPWMxSYNCI: Synchronization Input Pulse:** The value of the phase register is loaded into the counter register when an input synchronization pulse is detected (TBPHS → TBCTR). This operation occurs on the next valid time-base clock (TBCLK) edge.

  The delay from internal control module to target modules is given by:
  – if (TBCLK = MCPWMCLK): 2 x MCPWMCLK
  – if (TBCLK < MCPWMCLK): 1 x TBCLK
- **Software Forced Synchronization Pulse:** Writing a 1 to the TBCTL[SWFSYNC] control bit invokes a software forced synchronization. This pulse is ORed with the synchronization input signal, and therefore has the same effect as a pulse on MCPWMxSYNCI.

Note

If the MCPWMxSYNCI signal is held high, the sync does not continuously occur. The MCPWMxSYNCI is rising edge activated.

When modifying the TBPHS register during run-time, missed action qualifier events can occur due to sudden jumps in the TBCTR value at the time of the SYNCIN pulse. To recreate the behavior of missed action qualifier events, a CMPB event can be needed if CMPB is not already utilized.

This feature enables the MCPWM module to be automatically synchronized to the time base of another MCPWM module. Lead or lag phase control can be added to the waveforms generated by different MCPWM modules to synchronize them. In up-down-count mode, the TBCTL[PHSDIR] bit configures the direction of the time-base counter immediately after a synchronization event. The new direction is independent of the direction prior to the synchronization event. The PHSDIR bit is ignored in count-up or count-down modes (see Section 15.4.5 for examples).

Clearing the TBCTL[PHSEN] bit configures the MCPWM to ignore the synchronization input pulse.

15.4.3.4 MCPWM SYNC Selection

Table 15-3 specifies the sources for the MCPWM SYNC input.

Table 15-3. MCPWM SYNC Selection

PWM*.SYNCINSEL ECAP*.SYNCINSEL	Signal
0	Disable
1	INPUTXBAROUT5
2	INPUTXBAROUT6
3	ECAP1SYNCOUT
4	
5	PWM1SYNCOUT
6	
7	PWM3SYNCOUT
8	
9	
10	
11-31	Reserved

15.4.4 Phase Locking the Time-Base Clocks of Multiple MCPWM Modules

The TBCLKSYNC bit can be used to globally synchronize the time-base clocks of all enabled MCPWM modules on a device. This bit is part of the device's clock enable registers and is described in the *System Control and Interrupts* chapter. When TBCLKSYNC = 0, the time-base clock of all MCPWM modules is stopped (default). When TBCLKSYNC = 1, all MCPWM time-base clocks are started with the rising edge of TBCLK aligned. For synchronized TBCLKs, the prescaler bits in the TBCTL register of each MCPWM module must be set identically. The proper procedure for enabling the MCPWM clocks is:

1. Enable the individual MCPWM module clocks. This is described in the *System Control and Interrupts* chapter.
2. Set TBCLKSYNC = 0. This stops the time-base clock within any enabled MCPWM module.
3. Configure the prescaler values and desired MCPWM modes.
4. Set TBCLKSYNC = 1.

15.4.5 Time-Base Counter Modes and Timing Waveforms

The time-base counter operates in one of three modes:

- Up-count mode that is asymmetrical
- Up-down-count that is symmetrical
- Frozen where the time-base counter is held constant at the current value

To illustrate the operation of the first three modes, the following timing diagrams show when events are generated and how the time-base responds to an MCPWMxSYNCI signal.

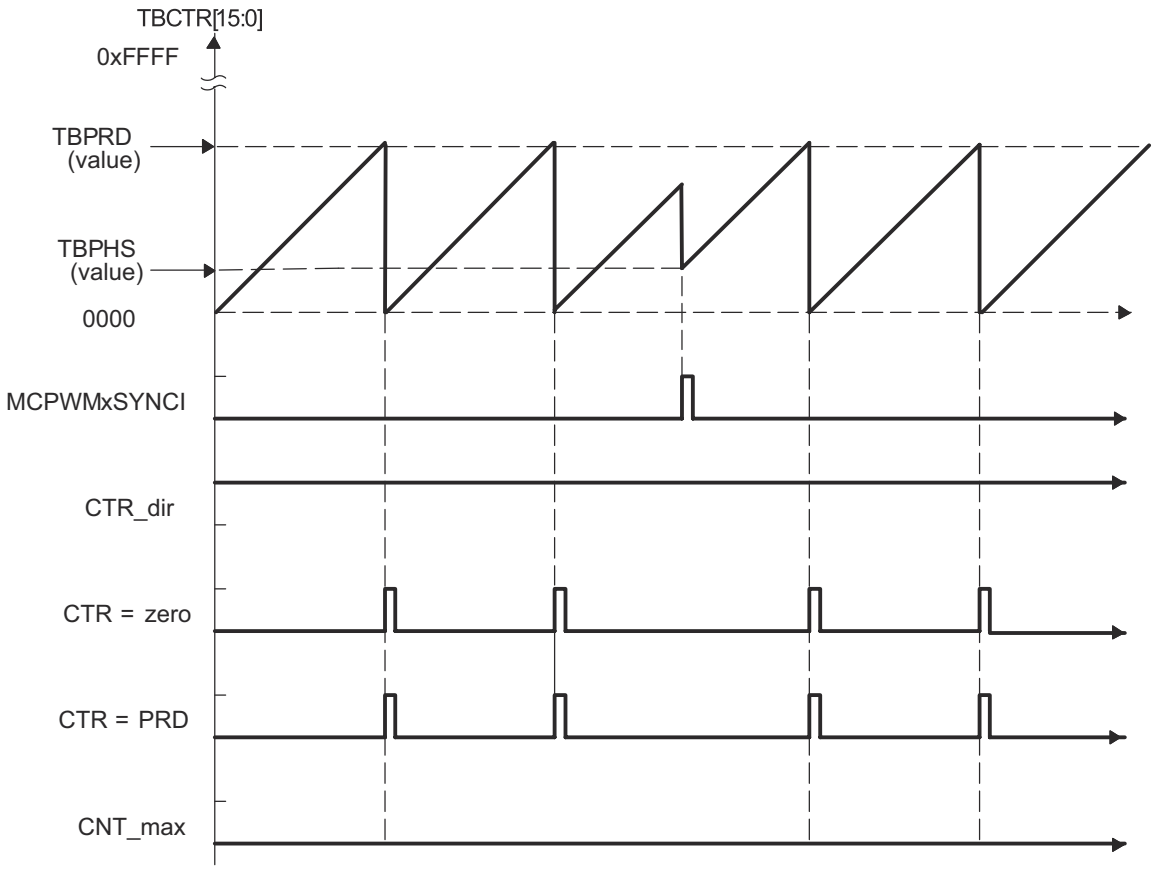


Figure 15-8. Time-Base Up-Count Mode Waveforms

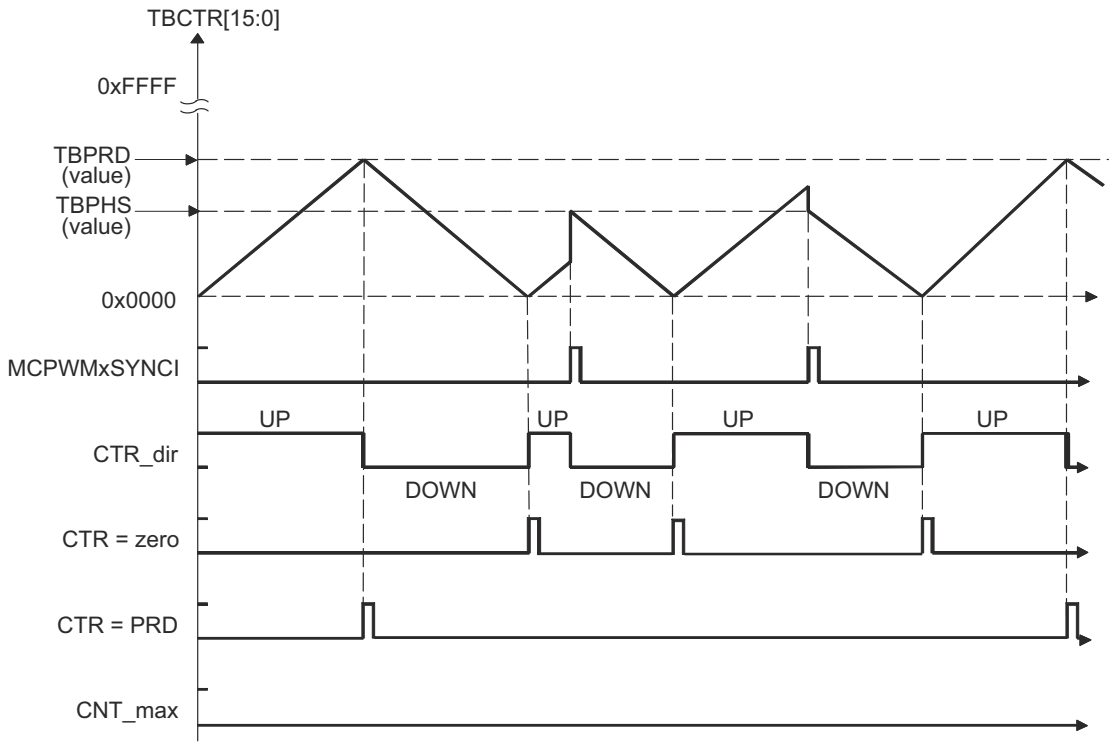


Figure 15-9. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

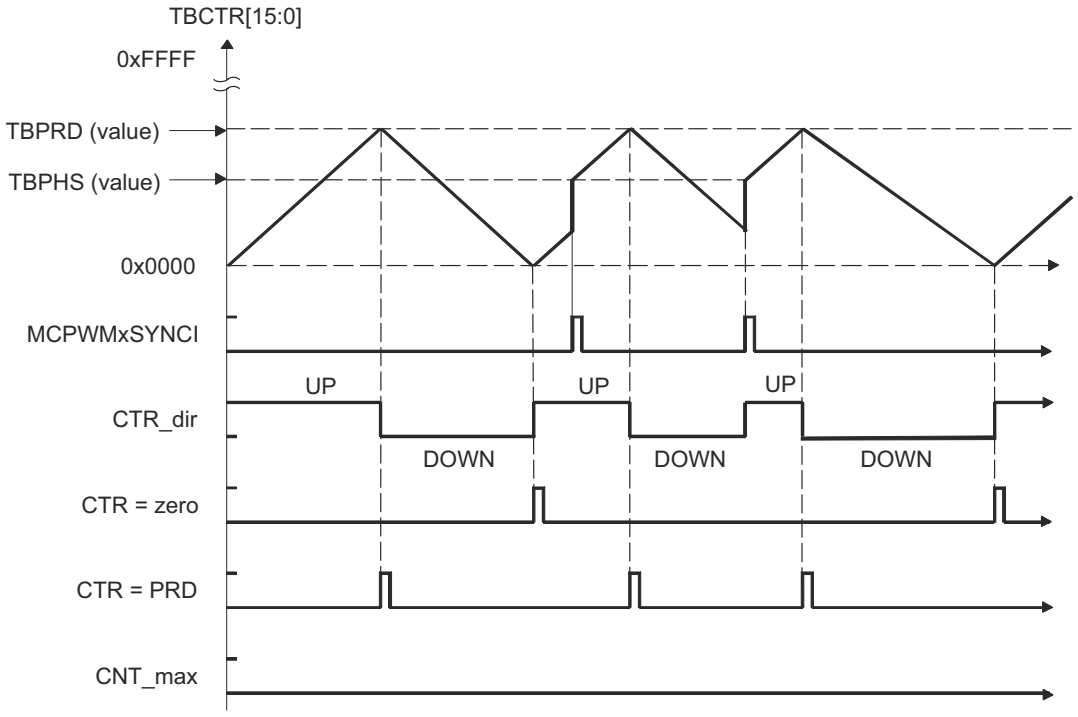


Figure 15-10. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up On Synchronization Event

15.4.6 Global Load

Figure 15-11 shows the signals and registers associated with the global load feature.

When this feature is enabled, the transfer of contents from the shadow register to the active register, for all registers that have a shadow register, occurs at the same event as defined by the configuration bits in Global Shadow to Active Load Control Register (GLDCTL[GLDMODE]). When GLDCTL[GLD] = 1, shadow to active load event selection bits for individual shadowed registers are ignored and global load mode takes effect for all registers with a corresponding shadow register.

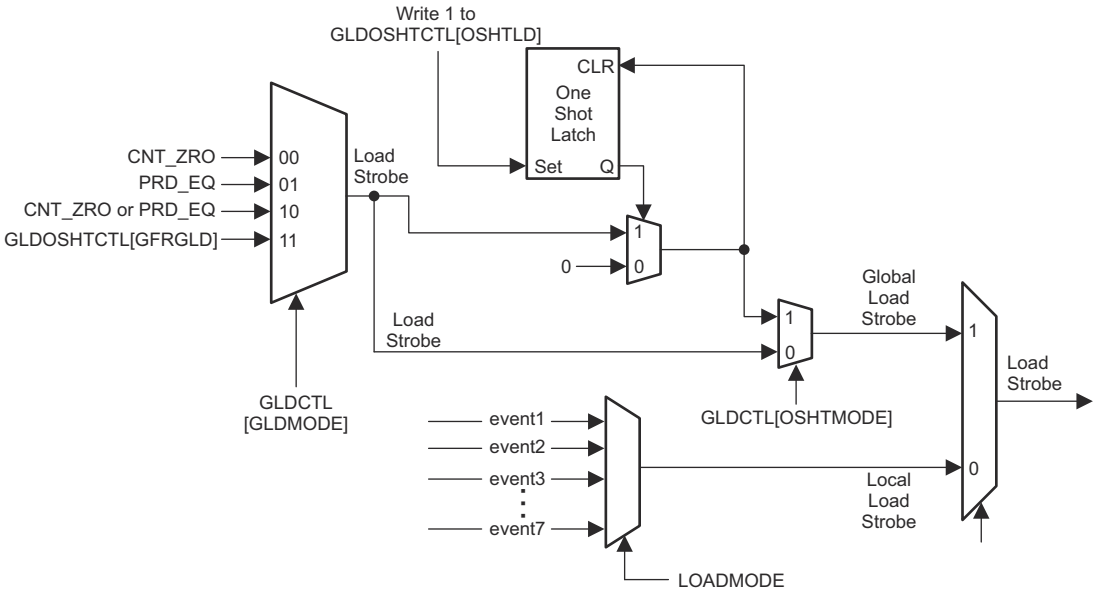


Figure 15-11. Global Load: Signals and Registers

15.4.6.1 One-Shot Load Mode

This feature causes the shadow-register to active-register transfers to occur once. When GLDOSHTCTL[OSHTLD] = 1 and one shot global load is enabled using GLDCTL[OSHTMODE], the shadow-register to active-register transfer takes place on the event selected by GLDCTL[GLDMODE]. Software force loading of contents from the shadow register to an active register is possible by using GLDOSHTCTL[GFRCLD].

15.5 Counter-Compare (CC) Submodule

Figure 15-12 illustrates the counter-compare submodule within the MCPWM.

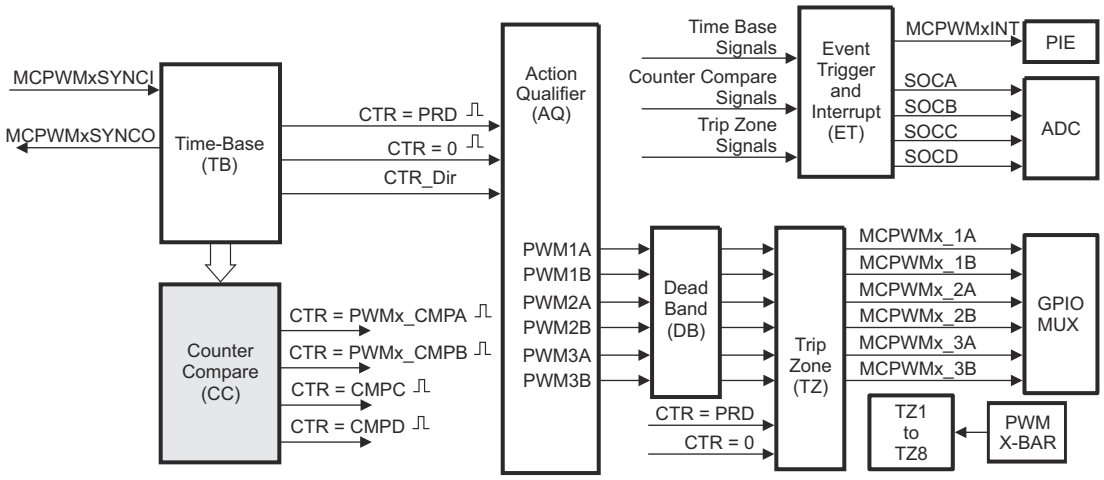


Figure 15-12. Counter-Compare Submodule

15.5.1 Purpose of the Counter-Compare Submodule

The counter-compare submodule takes as input the time-base counter value. This value is continuously compared to the counter-compare A (PWMx_CMPA), counter-compare B (PWMx_CMPB), counter-compare C (CMPC), and counter-compare D (CMPD) registers. When the time-base counter is equal to one of the compare registers, the counter-compare unit generates an appropriate event. Note that each MCPWM module can have up to three PWM pairs, denoted by "PWMx". There is a CMPA and CMPB register for each pair, but only one CMPC and CMPD register for the entire MCPWM module.

The counter-compare:

- Generates events based on programmable time stamps using the CMPA, CMPB, CMPC, and CMPD registers:
  - CTR = PWMx_CMPA: Time-base counter equals counter-compare A register for PWMx (TBCTR = PWMx_CMPA)
  - CTR = PWMx_CMPB: Time-base counter equals counter-compare B register (TBCTR = PWMx_CMPB)
  - CTR = PWMx_CMPC: Time-base counter equals counter-compare C register (TBCTR = CMPC)
  - CTR = PWMx_CMPD: Time-base counter equals counter-compare D register (TBCTR = CMPD)
- Controls the PWM duty cycle, if the action-qualifier submodule is configured appropriately using counter-compare A (PWMx_CMPA) and counter-compare B (PWMx_CMPB)
- Shadows new compare values to prevent corruption or glitches during the active PWM cycle

15.5.2 Controlling and Monitoring the Counter-Compare Submodule

The counter-compare submodule operation is shown in Figure 15-13.

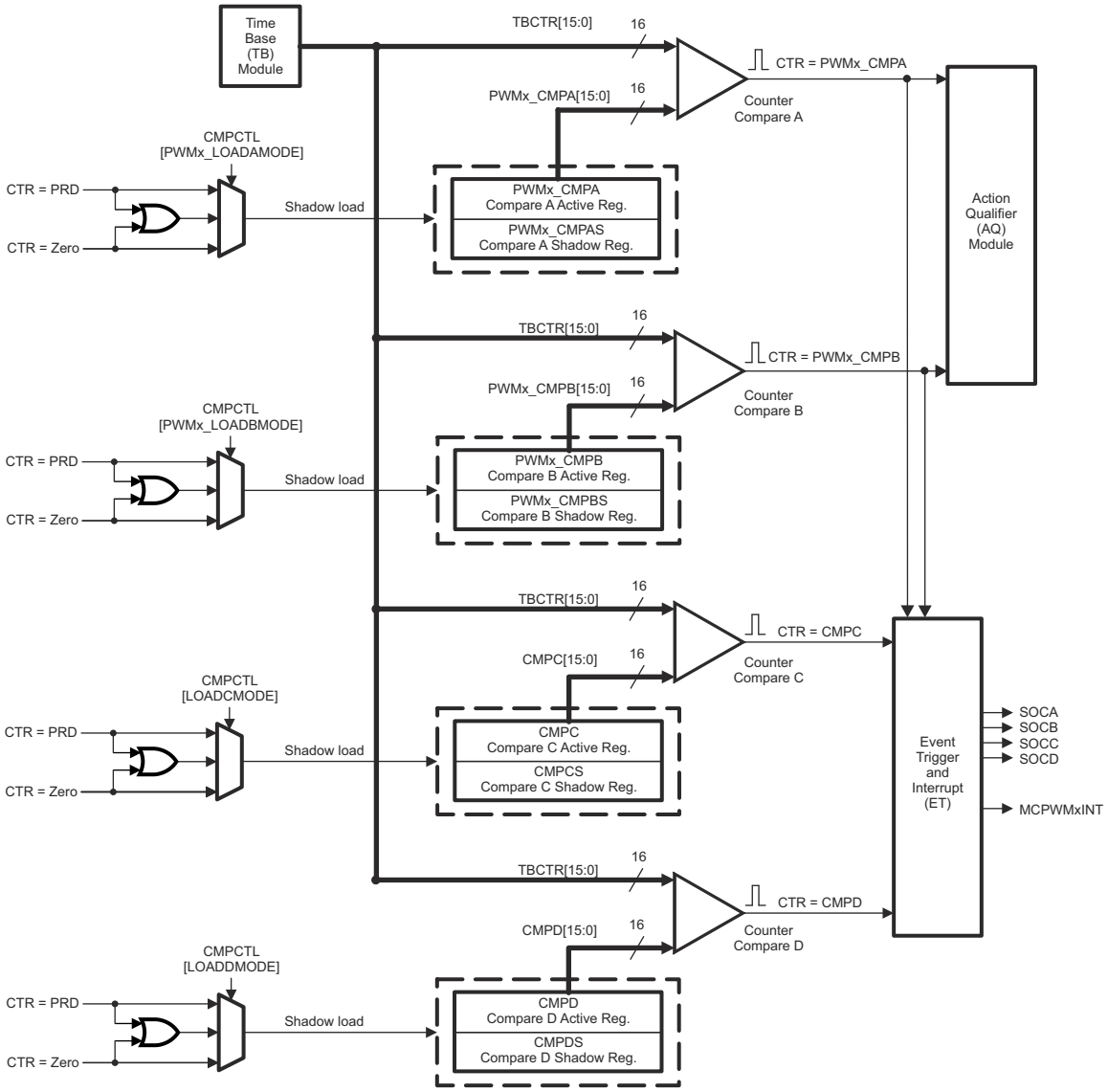


Figure 15-13. Detailed View of the Counter-Compare Submodule

15.5.3 Operational Highlights for the Counter-Compare Submodule

The counter-compare submodule is responsible for generating events that can be used in the action-qualifier and event-trigger submodules:

1. CTR = PWMx_CMPA: Time-base counter equal to counter-compare A register (TBCTR = PWMx_CMPA).
2. CTR = PWMx_CMPB: Time-base counter equal to counter-compare B register (TBCTR = PWMx_CMPB).
3. CTR = CMPC: Time-base counter equal to counter-compare C register (TBCTR = CMPC). This event can be used to generate an event in the event trigger submodule only.
4. CTR = CMPD: Time-base counter equal to counter-compare D register (TBCTR = CMPD). This event can be used to generate an event in the event trigger submodule only

For up-count mode, each event occurs only once per cycle. For up-down count mode, each event occurs twice per cycle if the compare value is between 0x00-TBPRD; and once per cycle if the compare value is equal to 0x00 or equal to TBPRD. These events are applied to the action-qualifier submodule where the events are qualified by the counter direction and converted into actions if enabled. Refer to Section 15.6.1 for more details.

The counter-compare registers PWMx_CMPA and PWMx_CMPB each have an associated shadow register. Shadowing provides a way to keep updates to the registers synchronized with the hardware. When shadowing is used, updates to the active registers only occur at strategic points. This prevents corruption or spurious operation due to the register being asynchronously modified by software. There is a separate memory address for the active and shadow registers. To utilize the shadow loading feature, the CPU writes to the corresponding shadow register instead of the active register. The behavior of the two load modes is:

Shadow Mode:

The shadow mode for the PWMx_CMPA and PWMx_CMPB is always enabled, however the user must write to the corresponding shadow register(PWMx_CMPAS or PWMx_CMPBS) for shadow loading to occur.

The content of the shadow register is transferred to the active register on one of the following events as specified by the CMPCTL[PWMx_LOADAMODE] and CMPCTL[PWMx_LOADBMODE] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD or CTR = Zero

Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.

Note

Refer to Section 15.6.5 for valid configurations of CMPA/CMPB and LOADAMODE/LOADBMODE.

Immediate Load Mode:

If the immediate load mode occurs when the CPU writes directly to the active register(PWMx_CMPA or PWMx_CMPB).

Additional Comparators

The counter-compare submodule is responsible for generating two additional independent compare events based on two compare registers, which is fed to Event Trigger submodule:

1. CTR = PWMx_CMPC: Time-base counter equal to counter-compare C register (TBCTR = CMPC).
2. CTR = PWMx_CMPD: Time-base counter equal to counter-compare D register (TBCTR = CMPD).

The counter-compare registers CMPC and CMPD each have an associated shadow register. The memory address of the active register and the shadow register are separate, similar to PWMx_CMPA and PWMx_CMPB. The value in the active CMPC and CMPD register is compared to the time-base counter (TBCTR). When the values are equal, the counter compare module generates an event. The shadow and active mode functionality is identical to the functionality for PWMx_CMPA and PWMx_CMPB described above.

Global Load Support

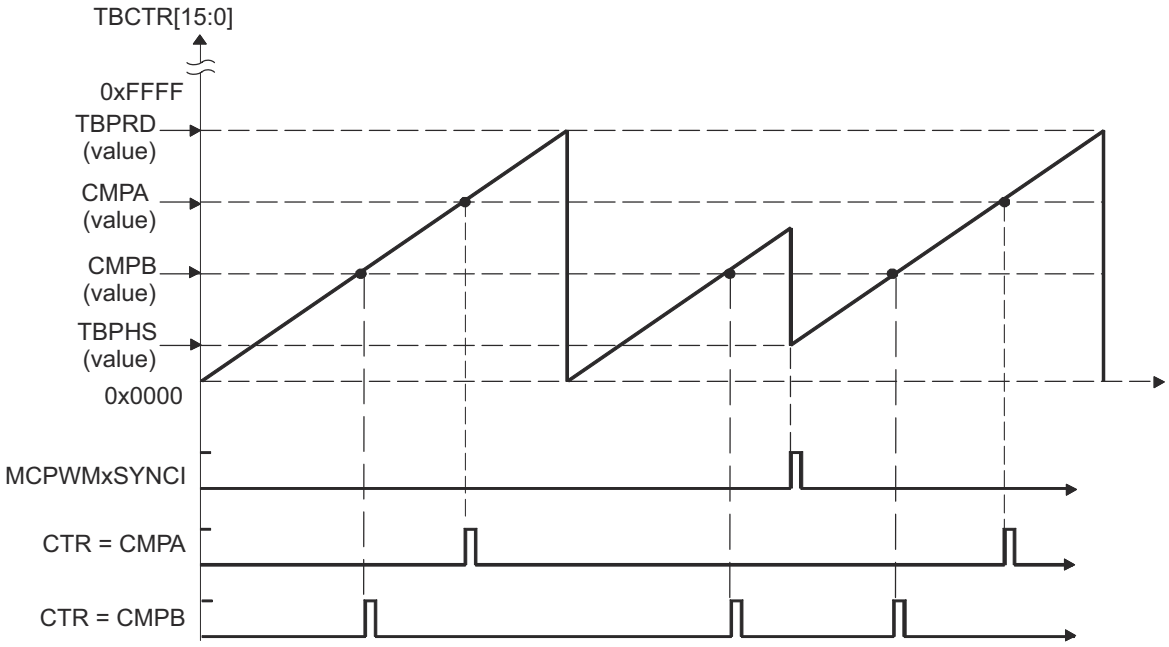
The global load control mechanism can also be used for all counter-compare registers by configuring the appropriate bits in the global load configuration register (GLDCTL). When the global load mode is selected the transfer of contents from shadow register to active register, for all registers that have a corresponding shadow register, occurs at the same event as defined by the configuration bits in the Global Shadow to Active Load Control Register (GLDCTL). The global load control mechanism is explained in Section 15.4.6.

15.5.4 Count Mode Timing Waveforms

The counter-compare module can generate compare events in both count modes:

- Up-count mode: used to generate an asymmetrical PWM waveform.
- Up-down-count mode: used to generate a symmetrical PWM waveform.

To best illustrate the operation of the two count modes, the timing diagrams in Figure 15-14 through Figure 15-16 show when events are generated and how the MCPWMxSYNCI signal interacts.



An MCPWMxSYNCI external synchronization event can cause a discontinuity in the TBCTR count sequence. This can lead to a compare event being skipped. This skipping is considered normal operation and must be taken into account.

Figure 15-14. Counter-Compare Event Waveforms in Up-Count Mode

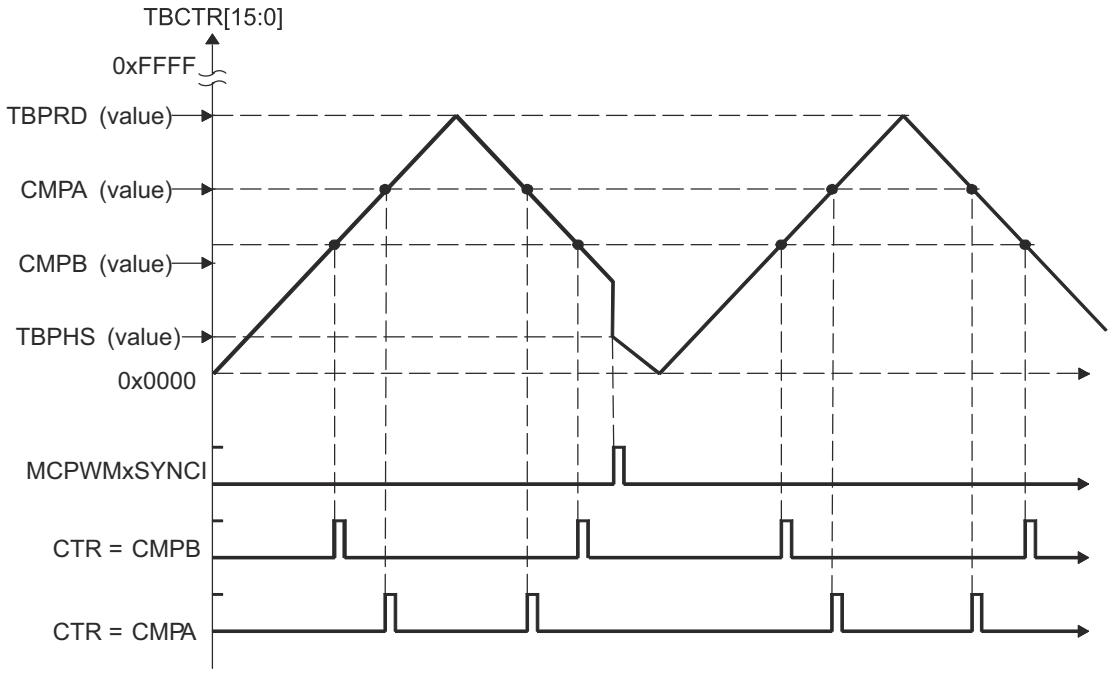


Figure 15-15. Counter-Compare Events In Up-Down Count Mode, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

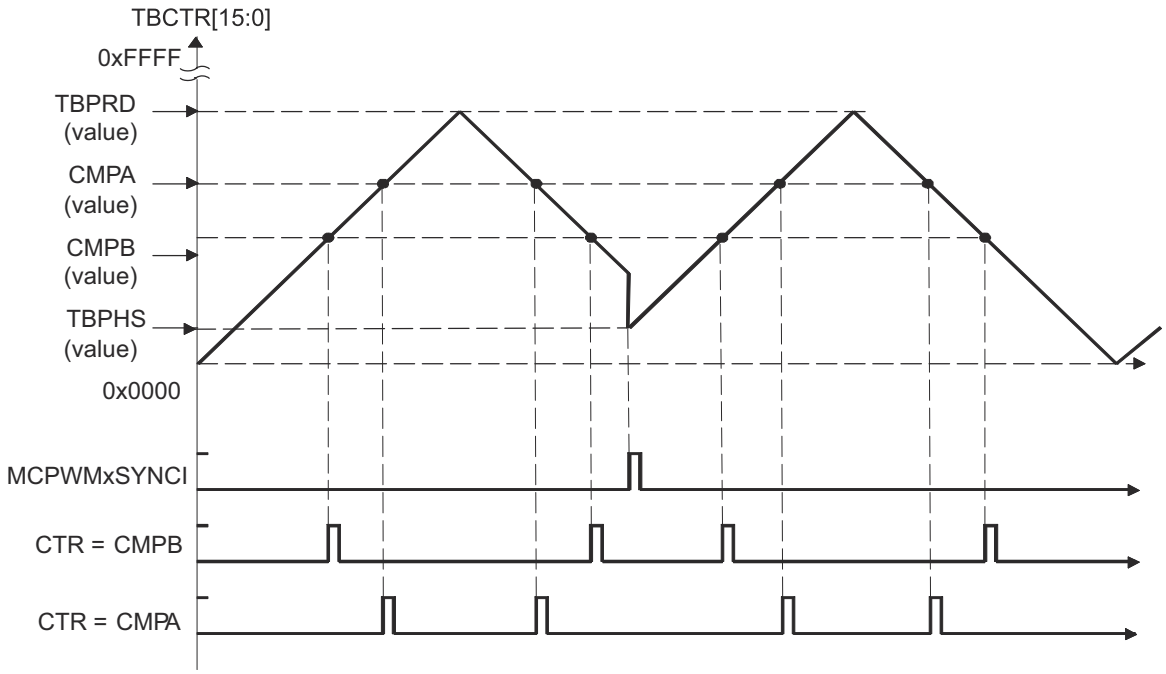


Figure 15-16. Counter-Compare Events In Up-Down Count Mode, TBCTL[PHSDIR = 1] Count Up On Synchronization Event

15.6 Action-Qualifier (AQ) Submodule

The action-qualifier submodule defines the behavior of MCPWMx_yA and MCPWMx_yB outputs for each MCPWM TBCTR event. This module can trigger a rising or falling edge on the PWM outputs.

Figure 15-17 illustrates the action-qualifier submodule within the MCPWM.

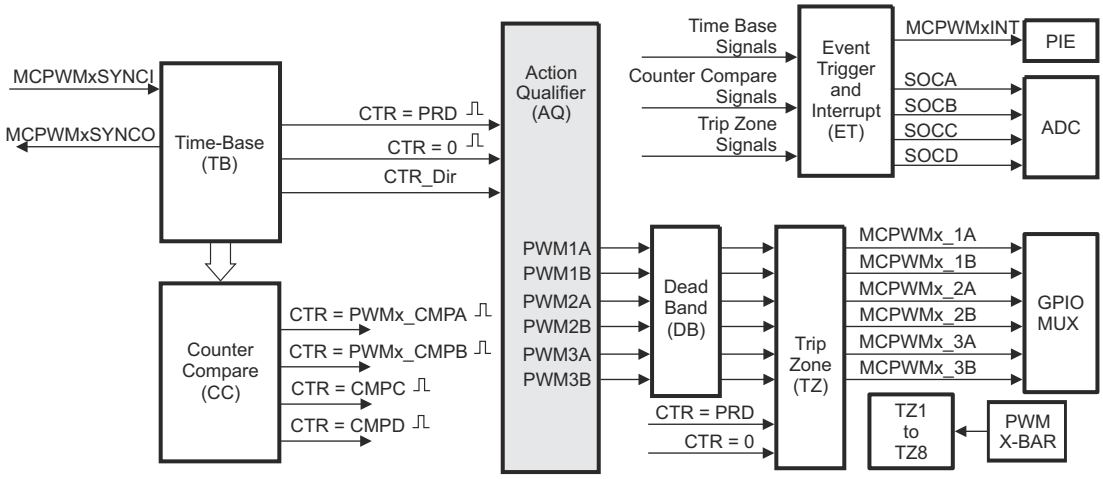


Figure 15-17. Action-Qualifier Submodule

15.6.1 Purpose of the Action-Qualifier Submodule

The action-qualifier submodule is responsible for the following:

- Qualifying and generating actions (set, clear, toggle) based on the following events:
  - CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
  - CTR = PWMx_CMPA: Time-base counter equal to the counter-compare A register (TBCTR = PWMx_CMPA)
  - CTR = PWMx_CMPB: Time-base counter equal to the counter-compare B register (TBCTR = PWMx_CMPB)
- Managing priority when these events occur concurrently
- Providing independent control of events when the time-base counter is increasing and when the time-base counter is decreasing

15.6.2 Action-Qualifier Submodule Control and Status Register Definitions

The action-qualifier submodule operation is shown in Figure 15-18 and monitored by way of the registers in Section 15.12.

For convenience, the possible input events are summarized again in Table 15-4.

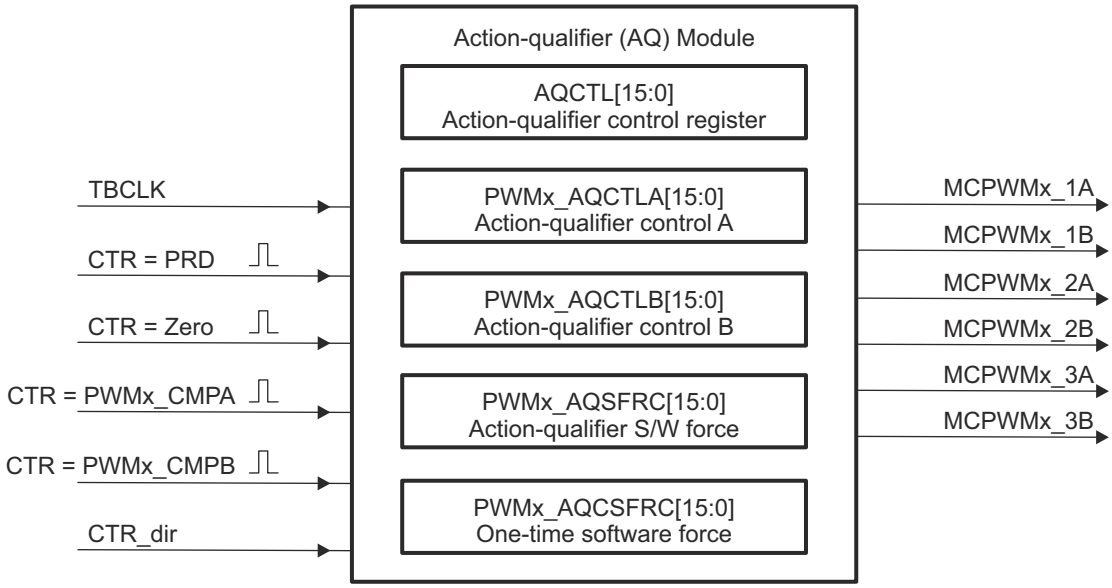


Figure 15-18. Action-Qualifier Submodule Inputs and Outputs

Table 15-4. Action-Qualifier Submodule Possible Input Events

Signal	Description	Registers Compared
CTR = PRD	Time-base counter equal to the period value	TBCTR = TBPRD
CTR = Zero	Time-base counter equal to 0	TBCTR = 0x00
CTR = CMPA	Time-base counter equal to the counter-compare A	TBCTR = PWMx_CMPA
CTR = CMPB	Time-base counter equal to the counter-compare B	TBCTR = PWMx_CMPB
Software forced event	Asynchronous event initiated by software	

The software forced action is a useful asynchronous event. This control is handled by the PWMx_AQSFRC and PWMx_AQOTSFRC register.

The action-qualifier submodule controls how the outputs MCPWMx_yA and MCPWMx_yB behave when a particular event occurs. The event inputs to the action-qualifier submodule are further qualified by the counter direction (up or down). This allows for independent action on outputs on both the count-up and count-down phases.

The possible actions imposed on outputs MCPWMx_yA and MCPWMx_yB are:

- **Set High:** Set output MCPWMx_yA or MCPWMx_yB to a high level.
- **Clear Low:** Set output MCPWMx_yA or MCPWMx_yB to a low level.
- **Toggle:** If MCPWMx_yA or MCPWMx_yB is currently pulled high, then pull the output low. If MCPWMx_yA or MCPWMx_yB is currently pulled low, then pull the output high.
- **Do Nothing:** Keep outputs MCPWMx_yA and MCPWMx_yB at same level as currently set. Although the "Do Nothing" option prevents an event from causing an action on the MCPWMx_yA and MCPWMx_yB outputs, this event can still trigger interrupts and ADC start of conversion. See the description in Section 15.9 for details.

Actions are specified independently for either output (MCPWMx_yA or MCPWMx_yB). Any or all events can be configured to generate actions on a given output. For example, both CTR = PWMx_CMPA and CTR = PWMx_CMPB can operate on output MCPWMx_yA. For each PWM pair "x", independent action qualifier events are specified. All qualifier actions are configured using the control registers found in Section 15.12.

For clarity, the illustrations in this chapter use a set of symbolic actions. These symbols are summarized in Figure 15-19. Each symbol represents an action as a marker in time. Some actions are fixed in time (zero and period) while the CMPA and CMPB actions are moveable and the time positions are programmed by way of the counter-compare A and B registers, respectively. To turn off or disable an action, use the "Do Nothing option"(the default at reset).

S/W force	TB Counter equals				Actions
	Zero	Comp A	Comp B	Period	
SW	Z	CA	CB	P	Do Nothing
SW	Z	CA	CB	P	Clear Lo
SW	Z	CA	CB	P	Set Hi
SW	Z	CA	CB	P	Toggle

Figure 15-19. Possible Action-Qualifier Actions for MCPWMx_yA and MCPWMx_yB Outputs

15.6.3 Action-Qualifier Event Priority

It is possible for the MCPWM action qualifier to receive more than one event at the same time. In this case, events are assigned a priority by the hardware. The general rule is events occurring later in time have a higher priority and software forced events always have the highest priority. The event priority levels for up-down count mode are shown in Table 15-5. A priority level of 1 is the highest priority and level 10 is the lowest. The priority changes slightly depending on the direction of TBCTR.

Table 15-5. Action-Qualifier Event Priority for Up-Down Count Mode

Priority Level	Event If TBCTR is Incrementing TBCTR = Zero up to TBCTR = TBPRD	Event If TBCTR is Decrementing TBCTR = TBPRD down to TBCTR = 1
1 (Highest)	Software forced event	Software forced event
2	Counter equals CMPB on up-count (CBU)	Counter equals CMPB on down-count (CBD)
3	Counter equals CMPA on up-count (CAU)	Counter equals CMPA on down-count (CAD)
4 (Lowest)	Counter equals zero	Counter equals period (TBPRD)

Table 15-6 shows the action-qualifier priority for up-count mode. In this case, the counter direction is always defined as up; therefore, down-count events never are taken.

Table 15-6. Action-Qualifier Event Priority for Up-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to period (TBPRD)
3	Counter equal to CMPB on up-count (CBU)
4	Counter equal to CMPA on up-count (CAU)
5 (Lowest)	Counter equal to Zero

It is possible to set the compare value greater than the period. In this case, the action takes place as shown in Table 15-7.

Table 15-7. Behavior if CMPA/CMPB is Greater than the Period

Counter Mode	Compare on Up-Count Event CAD/CBD	Compare on Down-Count Event CAD/CBD
Up-Count Mode	If CMPA/CMPB ≤ TBPRD period, then the event occurs on a compare match (TBCTR = CMPA or CMPB). If CMPA/CMPB > TBPRD, then the event does not occur.	Never occurs.
Up-Down Count Mode	If CMPA/CMPB < TBPRD and the counter is incrementing, the event occurs on a compare match (TBCTR = CMPA or CMPB). If CMPA/CMPB is ≥ TBPRD, the event occurs on a period match (TBCTR = TBPRD).	If CMPA/CMPB < TBPRD and the counter is decrementing, the event occurs on a compare match (TBCTR = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event occurs on a period match (TBCTR = TBPRD).

15.6.4 AQCTLA and AQCTLB Shadow Mode Operations

Shadow Mode:

Shadowing is always enabled for PWMx_AQCTLA and PWMx_AQCTLB. To utilize shadowing, the user must write to the corresponding shadow registers (PWMx_AQCTLAS and PWMx_AQCTLBS).

The content of the shadow register is transferred to the active register on one of the following events as specified by the AQCTL[PWMx_LDAQAMODE] and AQCTL[PWMx_LDAQBMODE] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD and CTR = Zero

Global Load Support

Global load control mechanism can also be used for PWMx_AQCTLA and PWMx_AQCTLB registers when global load is enabled via the global load configuration register (GLDCTL). When global load mode is selected, the transfer of contents from the shadow registers to active registers occurs at the same event as defined by the configuration bits in the Global Shadow to Active Load Control Register (GLDCTL). The global load control mechanism is explained in Section 15.4.6.

Immediate Load Mode:

To utilize immediate load mode, global load must be disabled and the user must write to the active register instead of the shadow register. See Figure 15-20 and Figure 15-21.

Note

Shadow to Active Load of Action Qualifier Output A/B Control Register [PWMx_AQCTLA and PWMx_AQCTLB] on CMPA = 0 or CMPB = 0 boundary

If the Counter-Compare A Register (PWMx_CMPA) or Counter-Compare B Register (PWMx_CMPB) is set to a value of 0 and the action qualifier action on PWMx_AQCTLA and PWMx_AQCTLB is configured to occur in the same instant as a shadow to active load (that is, CMPA = 0 and AQCTLA shadow to active load on TBCTR = 0 using AQCTL register LDAQAMODE and LDAQAMODE bits), then both events can enter contention depending on the timing of the write to PWMx_CMPA or PWMx_CMPB, resulting in a missed action qualifier event for one MCPWM period. It is recommended to use a Non-Zero Counter-Compare when using Shadow to Active Load of Action Qualifier Output A/B Control Register on TBCTR = 0 boundary.

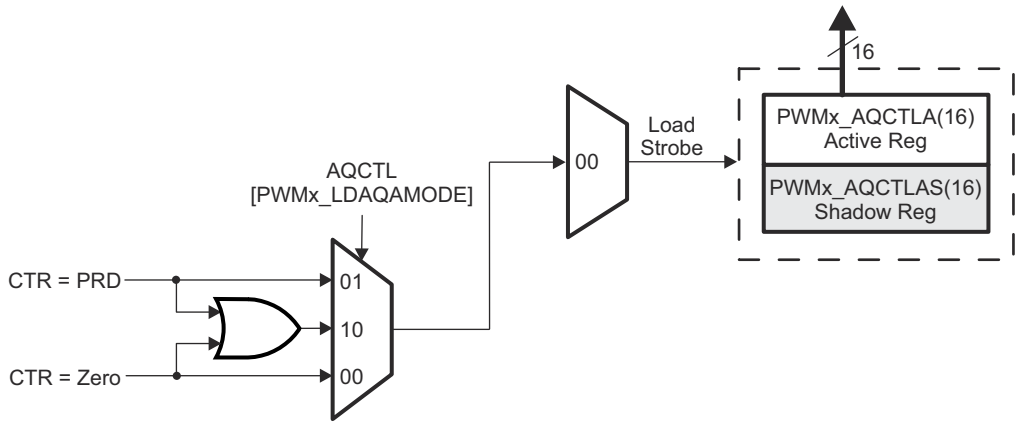


Figure 15-20. AQCTL[PWMx_LDAQAMODE]

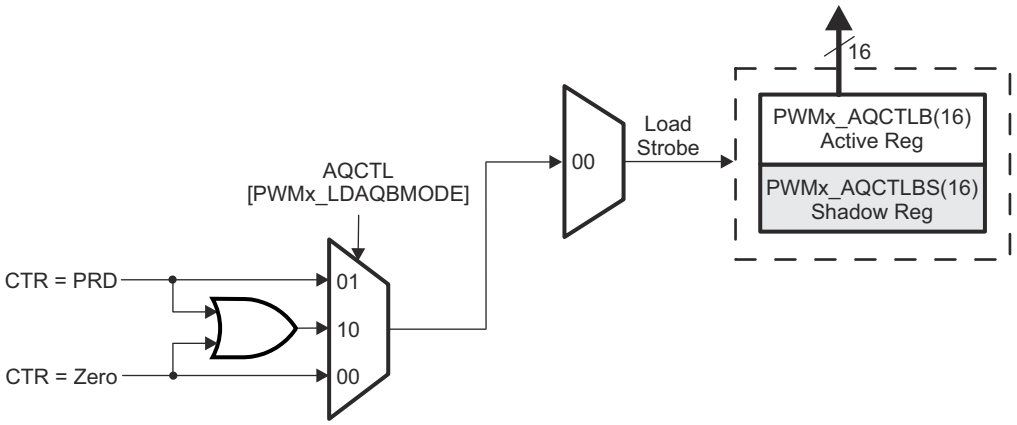


Figure 15-21. AQCTL[PWMx_LDAQBMODE]

15.6.5 Configuration Requirements for Common Waveforms

Note

The waveforms in this chapter show the behavior of the MCPWMs for a static compare register value. In a running system, the active compare registers (PWMx_CMPA and PWMx_CMPB) are typically updated from the respective shadow registers (PWMx_CMPAS and PWMx_CMPBS) once every period. Specify when the update takes place: either when the time-base counter reaches zero or when the time-base counter reaches the period. There are some cases when the action based on the new value can be delayed by one period or the action based on the old value can take effect for an extra period. Some PWM configurations avoid this situation. These include, but are not limited to, the following:

Use up-down count mode to generate a symmetric PWM:

- If loading PWMx_CMPA/PWMx_CMPB on zero, then use PWMx_CMPA/PWMx_CMPB values greater than or equal to 1.
- If loading PWMx_CMPA/PWMx_CMPB on period, then use PWMx_CMPA/PWMx_CMPB values less than or equal to TBPRD - 1.

  This means there is always a pulse of at least 1 TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Use up-down count mode to generate an asymmetric PWM:

- To achieve 50%-0% asymmetric PWM, use the following configuration: load PWMx_CMPA/PWMx_CMPB on period and use the period action to clear the PWM and a compare-up action to set the PWM. Modulate the compare value from 0 to TBPRD to achieve 50%-0% PWM duty.

When using up-count mode to generate an asymmetric PWM:

- To achieve 0-100% asymmetric PWM, load PWMx_CMPA/PWMx_CMPB on TBPRD. When PWMx_CMPA/PWMx_CMPB is not loaded on TBCTR = PRD, boundary conditions can occur depending on the timing of the write and the value written to PWMx_CMPA/PWMx_CMPB. Use the Zero action to set the PWM and a compare-up action to clear the PWM. Modulate the compare value from 0 to TBPRD + 1 to achieve 0-100% PWM duty.

When using up-count mode to generate an asymmetric PWM with dead-band enabled:

- To achieve 0%-100% PWM, use the following configuration: When the PWMx_CMPA value is too close to 0 or PRD such that the following conditions are met (CMPX < Deadband) or (CMPX > PRD – Deadband), the actions specified by the AQCTL register for CMPX do not take effect. To avoid this, the AQCTL settings must be altered under these conditions only to generate either high or low pulses for both CAU or CAD events (both set or both clear). Make sure that this software update is occurring synchronous to the PWM carrier cycle, and shadow mode is enabled.

When using up-down count mode to generate an asymmetric PWM with dead-band enabled:

- To achieve 0%-100% PWM, use the following configuration: When the PWMx_CMPA value is too close to 0 or PRD such that the following conditions are met (CMPX < Deadband/2) or (CMPX > PRD – (Deadband)/2), the actions specified by the AQCTL register for CMPX do not take effect. To avoid this, the AQCTL settings must be altered under these conditions only to generate either high or low pulses for both CAU or CAD events (both set or both clear). Make sure that this software update is occurring synchronous to the PWM carrier cycle, and shadow mode is enabled.

See *Using Enhanced Pulse Width Modulator (ePWM) Module for 0-100% Duty Cycle Control*.

Figure 15-22 shows how a symmetric PWM waveform can be generated using the up-down-count mode of the TBCTR. In this mode, 0%-100% DC modulation is achieved by using equal compare matches on the up count and down count portions of the waveform. In the example shown, PWMx_CMPA is used to make the comparison. When the counter is incrementing, the PWMx_CMPA match pulls the PWM output high. Likewise when the counter is decrementing, the compare match pulls the PWM signal low. When PWMx_CMPA = 0, the

PWM signal is high for the entire period giving a 100% duty waveform. When PWMx_CMPA = TBPRD, the PWM signal is low achieving 0% duty.

When using this configuration in practice, if loading PWMx_CMPA/PWMx_CMPB on zero, then use PWMx_CMPA/PWMx_CMPB values greater than or equal to 1. If loading PWMx_CMPA/PWMx_CMPB on period, then use PWMx_CMPA/PWMx_CMPB values less than or equal to TBPRD - 1. This means there is always a pulse of at least 1 TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

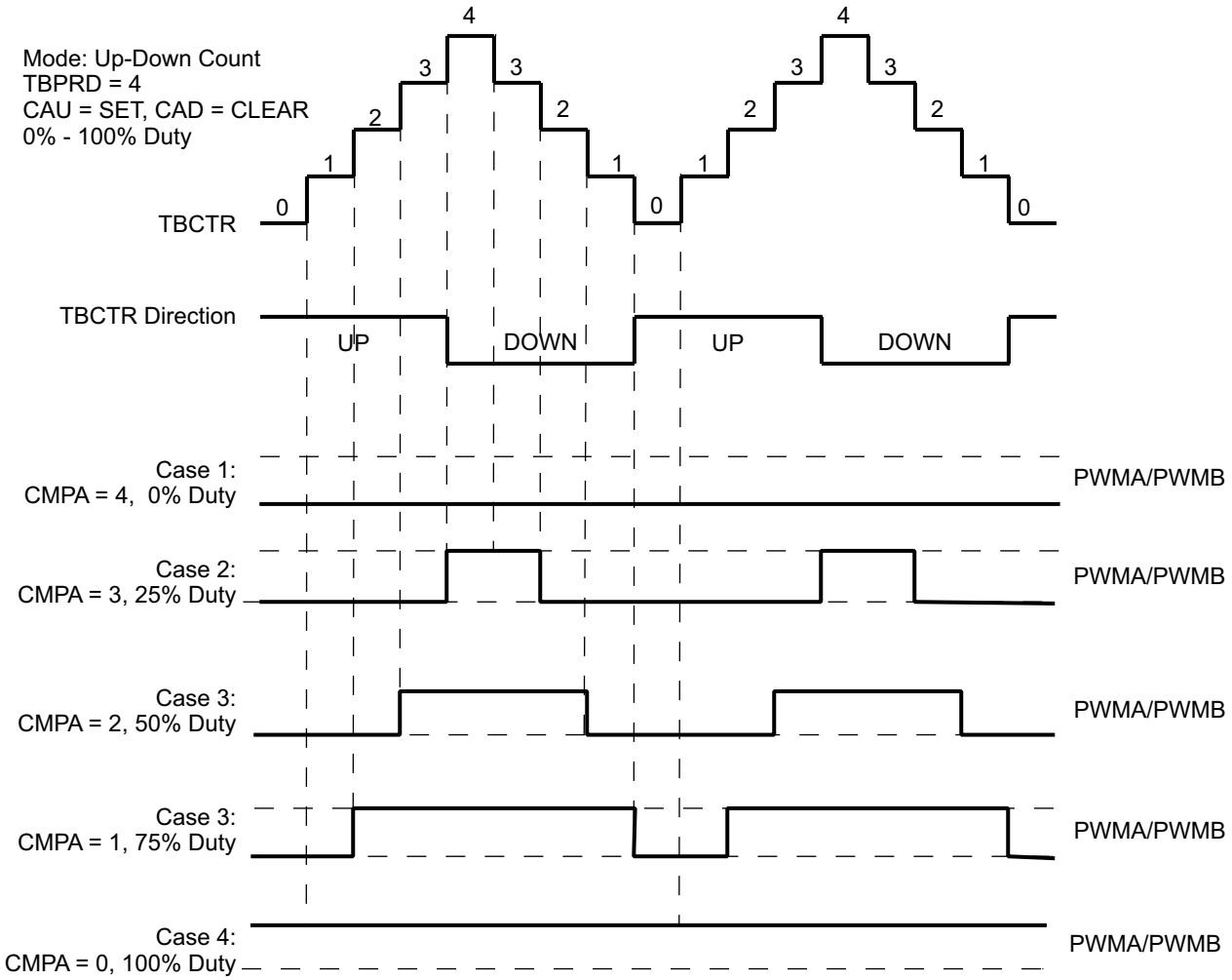
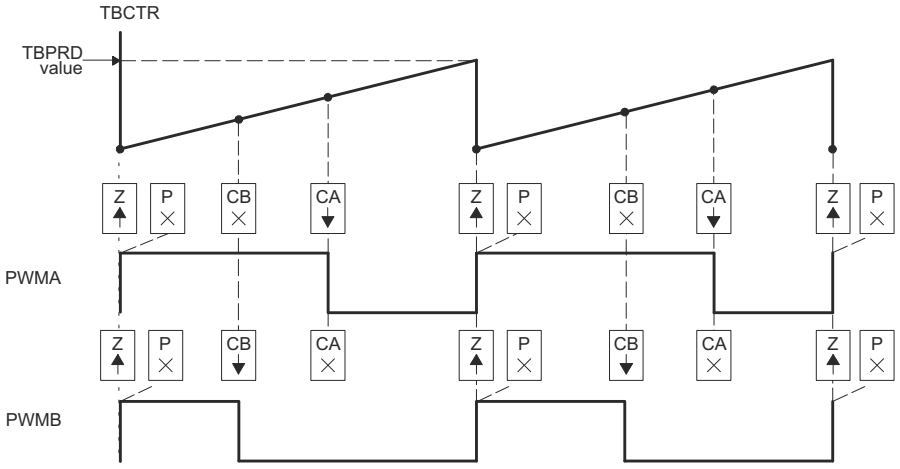


Figure 15-22. Up-Down Count Mode Symmetrical Waveform

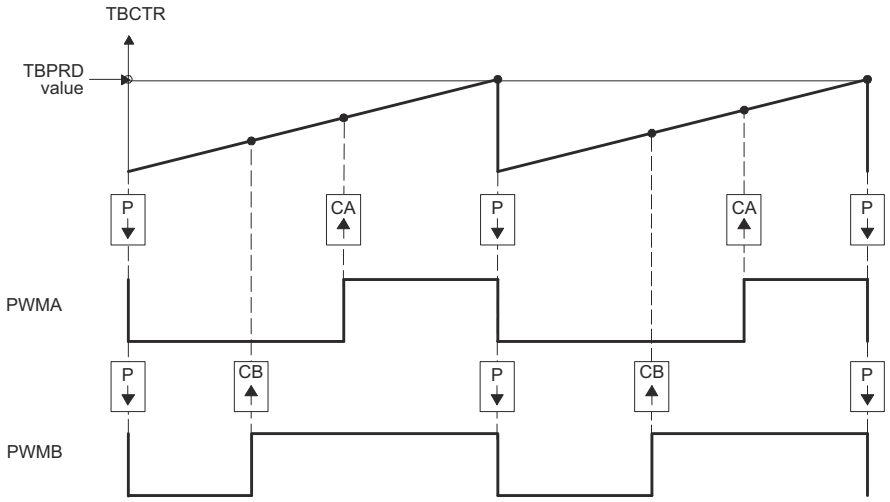
The following PWM waveforms show some common action-qualifier configurations. Some conventions used in the figures and examples are as follows:

- TBPRD, PWMx_CMPA, and PWMx_CMPB refer to the value written in the respective registers. The active register, not the shadow register, is used by the hardware.
- CMPx, refers to either PWMx_CMPA or PWMx_CMPB.
- MCPWMx_yA and MCPWMx_yB refer to the output signals from MCPWMx, Channel pair y.
- Up-Down means count-up and count-down mode, Up means up-count mode.
- Sym = Symmetric, Asym = Asymmetric.



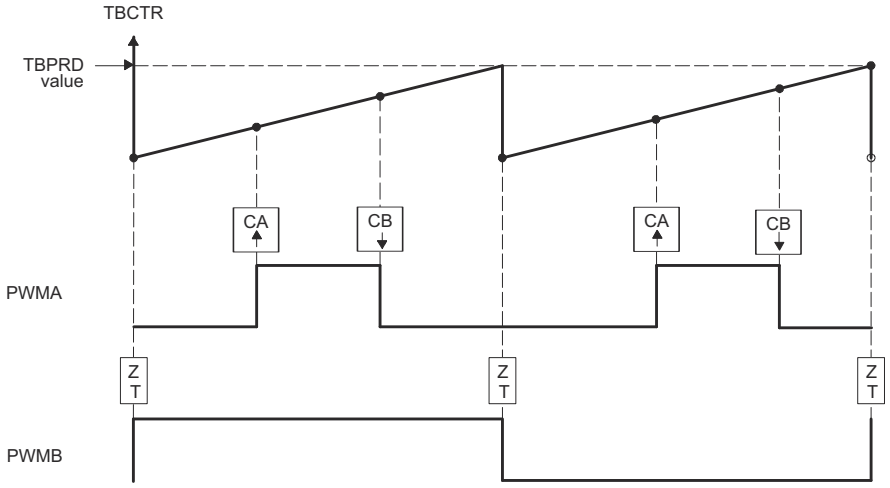
A. PWM period = (TBPRD + 1) × T_{TBCLK}
B. Duty modulation for MCPWMx_yA is set by PWMy_CMPA, and is active high (that is, high time duty proportional to PWMy_CMPA).
C. Duty modulation for MCPWMx_yB is set by PWMy_CMPB and is active high (that is, high time duty proportional to PWMy_CMPB).
D. The "Do Nothing" actions (X) are shown for completeness, but are not shown on subsequent diagrams.
E. Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

Figure 15-23. Up, Single Edge Asymmetric Waveform, with Independent Modulation on PWMxA and PWMxB—Active High



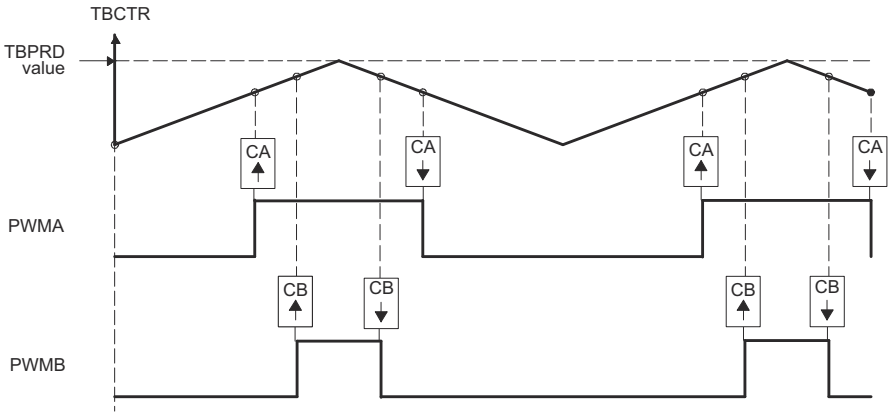
A. PWM period = (TBPRD + 1) × T_{TBCLK}
B. Duty modulation for MCPWMx_yA is set by PWMy_CMPA, and is active low (that is, the low time duty is proportional to PWMy_CMPA).
C. Duty modulation for MCPWMx_yB is set by PWMy_CMPB and is active low (that is, the low time duty is proportional to PWMy_CMPB).
D. Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

Figure 15-24. Up, Single Edge Asymmetric Waveform with Independent Modulation on PWMxA and PWMxB—Active Low



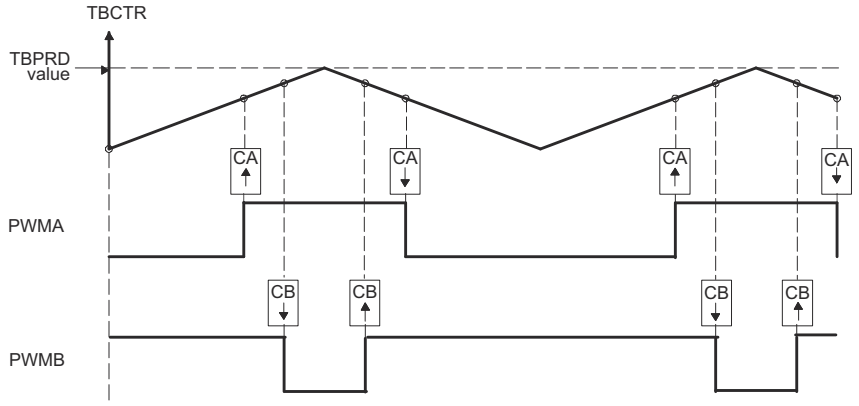
A. PWM frequency = 1/((TBPRD + 1) × T_{TBCLK})
B. Pulse can be placed anywhere within the PWM cycle (0000 - TBPRD)
C. High time duty proportional to (CMPB - CMPA)

Figure 15-25. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on PWMxA



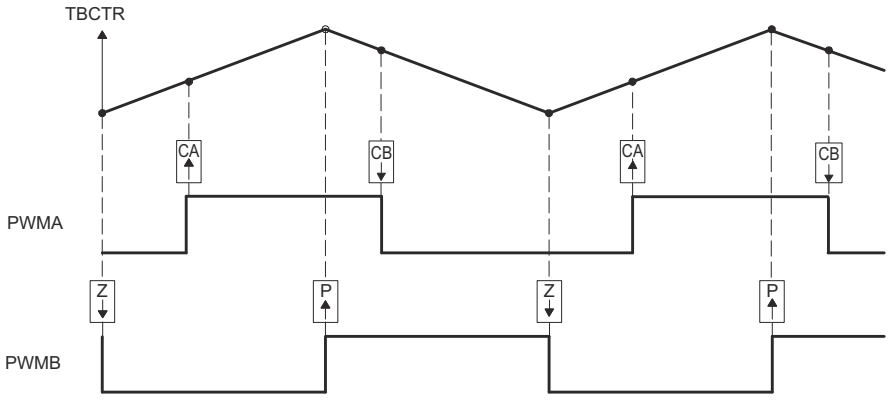
A. PWM period = 2 x TBPRD × T_{TBCLK}
B. Duty modulation for MCPWMx_yA is set by PWMy_CMPA, and is active low (that is, the low time duty is proportional to PWMy_CMPA).
C. Duty modulation for MCPWMx_yB is set by PWMy_CMPB and is active low (that is, the low time duty is proportional to PWMy_CMPB).
D. Outputs MCPWMx_yA and MCPWMx_yB can drive independent power switches.

Figure 15-26. Up-Down Count, Dual-Edge Symmetric Waveform, with Independent Modulation on PWMxA and PWMxB — Active Low



A. PWM period = 2 × TBPRD × T_{TBCLK}
B. Duty modulation for MCPWMx_yA is set by PWMy_CMPA, and is active low, that is, low time duty proportional to PWMy_CMPA.
C. Duty modulation for MCPWMx_yB is set by PWMy_CMPB and is active high, that is, high time duty proportional to PWMy_CMPB.
D. Outputs MCPWMx can drive upper/lower (complementary) power switches.
E. Dead-band = CMPB - CMPA (fully programmable edge placement by software). Note the dead-band module is also available if the more classical edge delay method is required.

Figure 15-27. Up-Down Count, Dual-Edge Symmetric Waveform, with Independent Modulation on PWMxA and PWMxB — Complementary



A. PWM period = 2 × TBPRD × TBCLK
B. Rising edge and falling edge can be asymmetrically positioned within a PWM cycle. This allows for pulse placement techniques.
C. Duty modulation for MCPWMx_yA is set by PWMy_CMPA and PWMy_CMPB.
D. Low time duty for MCPWMx_yA is proportional to (PWMy_CMPA + PWMy_CMPB).
E. To change this example to active high, PWMy_CMPA and PWMy_CMPB actions need to be inverted (that is, Clear on PWMy_CMPA, Set on PWMy_CMPB).
F. Duty modulation for MCPWMx_yB is fixed at 50% (utilizes spare action resources for MCPWMx_yB).

Figure 15-28. Up-Down Count, Dual-Edge Asymmetric Waveform, with Independent Modulation on PWMxA—Active Low

15.7 Dead-Band Generator (DB) Submodule

Figure 15-29 illustrates the dead-band submodule within the MCPWM.

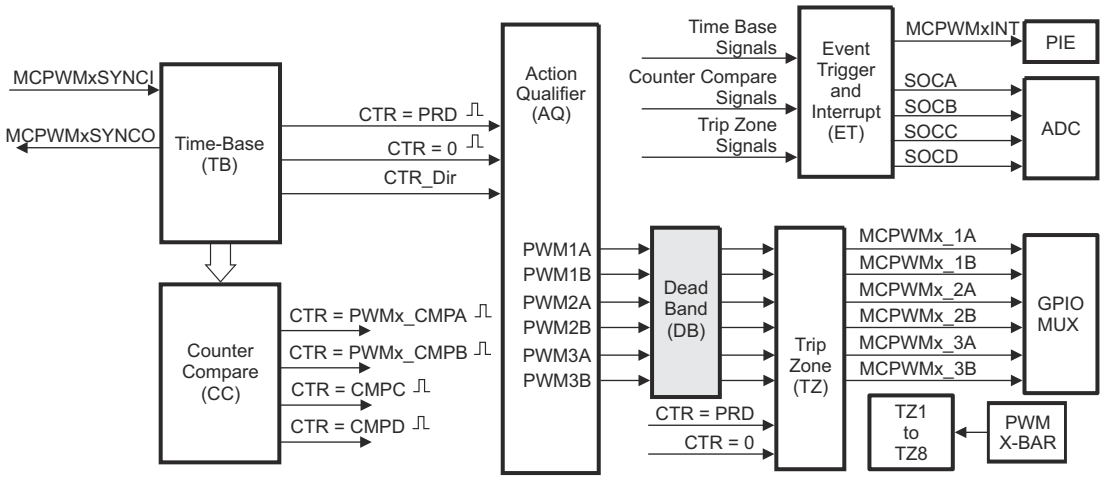


Figure 15-29. Dead_Band Submodule

15.7.1 Purpose of the Dead-Band Submodule

The action-qualifier (AQ) module section discussed how the AQ module can generate the required dead band by having full control over edge placement using both the PWMx_CMPA and PWMx_CMPB resources of the MCPWM module. However, if the more classical edge delay-based dead band with polarity control is required, then the dead-band submodule described here must be used.

The key functions of the dead-band module are:

- Generating appropriate signal pairs (MCPWMx_yA and MCPWMx_yB) with dead-band relationship from a single MCPWMx_yA input
- Programming signal pairs for:
  - Active high (AH)
  - Active low (AL)
  - Active high complementary (AHC)
  - Active low complementary (ALC)
- Adding programmable delay to rising edges (RED)
- Adding programmable delay to falling edges (FED)
- Can be totally bypassed from the signal path (note dotted lines in diagram)

15.7.2 Dead-Band Submodule Additional Operating Modes

On the type-1 ePWM, RED can appear on one channel output and FED can appear on the other channel output.

The dead-band module for MCPWM is unchanged from a type-4 ePWM with the following exceptions:

- Addition of memory-mapped shadow registers for DBCTL, DBFED, and DBRED.
- High-resolution dead-band and half-cycle clocking mode are removed.
- The shadow register for DBCTL is removed.

The following list shows the distinct difference between type-1 and type-4 ePWM modules with respect to dead-band operating modes:

- By adding S6, S7, and S8 in Figure 15-30, RED and FED can appear on both the A-channel and B-channel outputs. Additionally, both RED and FED together can be applied to either the A-channel or B-channel outputs to allow B-channel phase shifting with respect to the A-channel.

Note

Phase shifting B-channel with respect to the A-channel using the dead-band submodule additional operating modes has limitations with respect to the choice of RED and FED delay with respect to the operating duty cycle of the PWMxA and PWMxB outputs.

- The dead-band counters have also been increased to 14 bits
- Dead-band and dead-band high-resolution registers are now shadowed

Note

Cannot have both RED and FED together applied to both PWMxA and PWMxB. RED and FED together can be applied only to either OutA OR OutB.

Phase shifting B-channel with respect to the A-channel: When PWMxB is derived from PWMxA using the DEDB_MODE bit and by delaying rising edge and falling edge by the phase shift amount. When the duty cycle value on PWMxA is less than this phase shift amount, PWMxA's falling edge has precedence over the delayed rising edge for PWMxB. Make sure the duty cycle value of the current waveform applied to the dead-band module is greater than the required phase shift amount.

The type-4 action qualifier and dead-band outputs of the ePWM module are delayed by one TBCLK cycle in comparison to the type-2 ePWM module, although the type-4 ePWM behavior is the same as the type-3 ePWM. Both PWMA and PWMB signals are delayed under all circumstances.

Shadow Mode:

The shadow mode for the DBRED and DBFED registers is always enabled; however, write to the memory-mapped shadow registers (DBREDS, DBFEDS) to utilize shadow loading. Writing to the active register results in an immediate load.

If writing to the shadow register, the content of the shadow register is transferred to the active register on one of the following events as specified by the DBCTL [LOADREDMODE] and DBCTL [LOADFEDMODE] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD and CTR = Zero

Global Load Support

Global load control mechanism can also be used for DBRED and DBFED registers by configuring the appropriate bits in the global load configuration register (GLDCTL). When global load mode is selected the transfer of contents from shadow register to active register, for all registers that have a shadow register, occurs at the same event as defined by the configuration bits in the Global Shadow to Active Load Control Register (GLDCTL). The Global load control mechanism is explained in Section 15.4.6.

Note

When DBRED/DBFED active is loaded with a new shadow value while DB counters are counting, the new DBRED/DBFED value only affects the NEXT PWMx edge and not the current edge.

A dead-band value of zero cannot be used when the Global Shadow to Active Load is set to occur at CTR = ZERO. Similarly, a dead-band value of PRD cannot be used when the Global Shadow to Active Load is set to occur at CTR = PRD.

15.7.3 Operational Highlights for the Dead-Band Submodule

The configuration options for the dead-band submodule are shown in Figure 15-30.

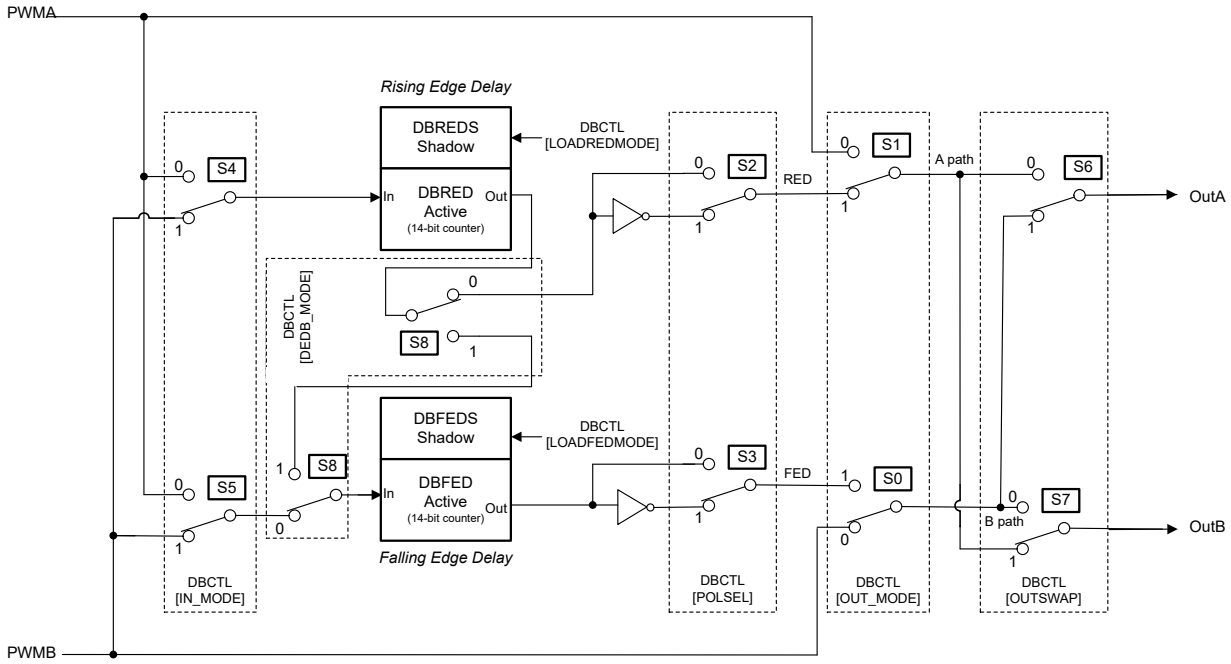


Figure 15-30. Configuration Options for the Dead-Band Submodule

Note

Note that the dead-band settings for all PWM pairs are configured by the same register bits, it is not possible to have separate dead-band settings for separate PWM pairs within a single MCPWM module.

Although all combinations are supported, not all are typical usage modes. Table 15-8 documents some classical dead-band configurations. These modes assume that the DBCTL[IN_MODE] is configured such that MCPWMx_yA In is the source for both falling-edge and rising-edge delay. Enhanced, or non-traditional modes can be achieved by changing the input signal source. The modes shown in Table 15-8 fall into the following categories:

- **Mode 1: Bypass both falling-edge delay (FED) and rising-edge delay (RED):** Allows the user to fully disable the dead-band submodule from the PWM signal path.
- **Mode 2-5: Classical Dead-Band Polarity Settings:** These represent typical polarity configurations that can address all the active-high and active-low modes required by available industry power switch gate drivers. The waveforms for these typical cases are shown in Figure 15-31. Note that to generate equivalent waveforms to Figure 15-31, configure the action-qualifier submodule to generate the signal as shown for MCPWMx_yA.
- **Mode 6: Bypass rising-edge delay (RED) and Mode 7: Bypass falling-edge delay (FED):** Finally the last two entries in Table 15-8 show combinations where either the falling-edge delay (FED) or rising-edge delay (RED) blocks are bypassed.

Figure 15-31 shows waveforms for typical cases where 0% < duty < 100%.

Table 15-8. Classical Dead-Band Operating Modes

Mode	Mode Description	DBCTL[POLSEL]		DBCTL[OUT_MODE]	
		S3	S2	S1	S0
1	MCPWMx_yA and MCPWMx_yB Passed Through (No Delay)	X	X	0	0
2	Active High Complementary (AHC)	1	0	1	1
3	Active Low Complementary (ALC)	0	1	1	1
4	Active High (AH)	0	0	1	1
5	Active Low (AL)	1	1	1	1
6	MCPWMx_yA Out = MCPWMx_yA In (No Delay) MCPWMx_yB Out = MCPWMx_yA In with Falling-Edge Delay	0 or 1	0 or 1	0	1
7	MCPWMx_yA Out = MCPWMx_yA In with Rising-Edge Delay MCPWMx_yB Out = MCPWMx_yB In with No Delay	0 or 1	0 or 1	1	0

Table 15-9. Additional Dead-Band Operating Modes

Mode Description	DBCTL[DEDB-MODE]	DBCTL[OUTSWAP]	
	S8	S6	S7
MCPWMx_yA and MCPWMx_yB signals are as defined by OUT-MODE bits.	0	0	0
MCPWMx_yA = A-path as defined by OUT-MODE bits. MCPWMx_yB = A-path as defined by OUT-MODE bits (rising-edge delay or delay-bypassed A-signal path)	0	0	1
MCPWMx_yA = B-path as defined by OUT-MODE bits (falling-edge delay or delay-bypassed B-signal path) MCPWMx_yB = B-path as defined by OUT-MODE bits	0	1	0
MCPWMx_yA = B-path as defined by OUT-MODE bits (falling-edge delay or delay-bypassed B-signal path) MCPWMx_yB = A-path as defined by OUT-MODE bits (rising-edge delay or delay-bypassed A-signal path)	0	1	1
Rising-edge delay applied to MCPWMx_yA / MCPWMx_yB as selected by S4 switch (IN-MODE bits) on A signal path only. Falling-edge delay applied to MCPWMx_yA / MCPWMx_yB as selected by S5 switch (IN-MODE bits) on B signal path only.	0	X	X
Rising-edge delay and falling-edge delay applied to source selected by S4 switch (IN-MODE bits) and output to B signal path only.(1)	1	X	X

(1) When this bit is set to 1, the user can always either set OUT_MODE bits such that Apath = InA or set OUTSWAP bits such that MCPWMx_yA=Bpath. Otherwise, MCPWMx_yA is invalid.

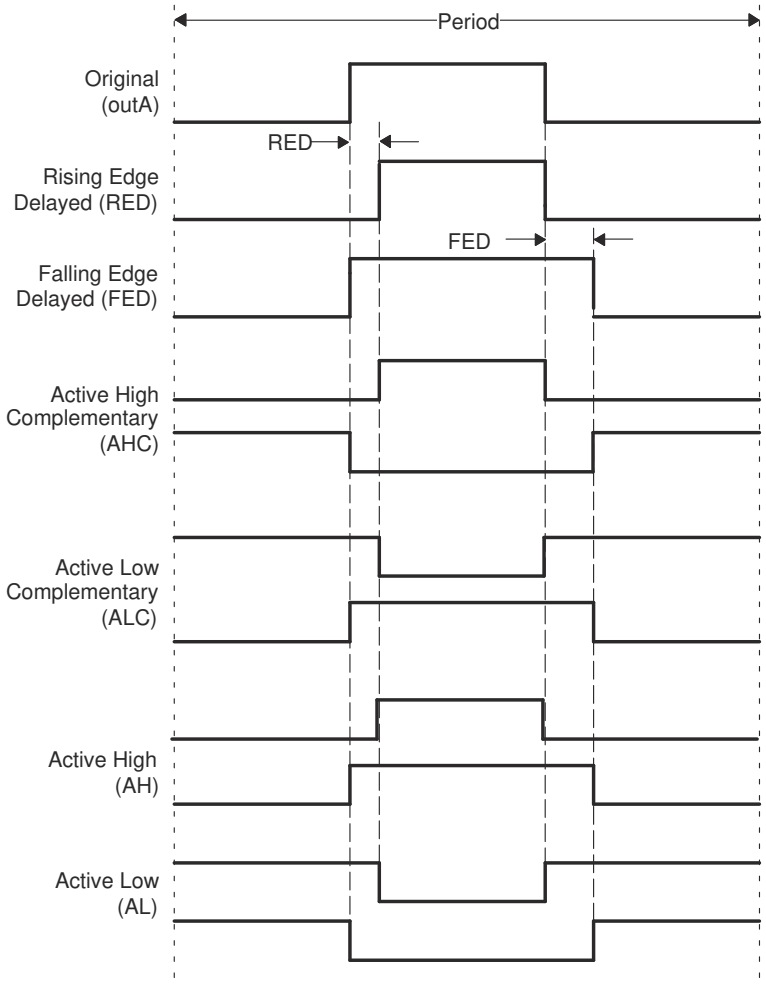


Figure 15-31. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)

The dead-band submodule supports independent values for rising-edge (RED) and falling-edge (FED) delays. The amount of delay is programmed using the DBRED and DBFED registers. These are 10-bit registers and the value represents the number of TBCLK (time-base clock) pulses by which a signal edge is delayed. For example, the formula to calculate falling-edge-delay and rising-edge-delay is:

$FED = DBFED \times T_{TBCLK}$

$RED = DBRED \times T_{TBCLK}$

Where T_{TBCLK} is the period of TBCLK, the prescaled version of MCPWMCLK.

For convenience, delay values for various TBCLK options are shown in Table 15-10. The MCPWM input clock frequency that these delay values been computed by is 100MHz.

Table 15-10. Dead-Band Delay Values in µs as a Function of DBFED and DBRED

Dead-Band Value	Dead-Band Delay (µs)		
DBFED, DBRED	TBCLK = PWMCLK/1	TBCLK = PWMCLK /2	TBCLK = PWMCLK/4
1	0.01	0.02	0.04
5	0.05	0.10	0.20
10	0.10	0.20	0.40
100	1.00	2.00	4.00
200	2.00	4.00	8.00
400	4.00	8.00	16.00
500	5.00	10.00	20.00
600	6.00	12.00	24.00
700	7.00	14.00	28.00
800	8.00	16.00	32.00
900	9.00	18.00	36.00
1000	10.00	20.00	40.00

15.8 Trip-Zone (TZ) Submodule

Each MCPWM module is connected to eight $\overline{TZn}$ signals ($\overline{TZ1}$ to $\overline{TZ8}$). All eight $\overline{TZn}$ signals are sourced from the PWM X-BAR that is inherited from a type-4 ePWM. Refer to the *Crossbar (X-BAR)* chapter to see what signals can be routed to the MCPWM module to be used as a trip source. These signals indicate external fault or trip conditions, and the MCPWM outputs can be programmed to respond accordingly when faults occur.

Figure 15-32 illustrates the trip-zone submodule within the MCPWM.

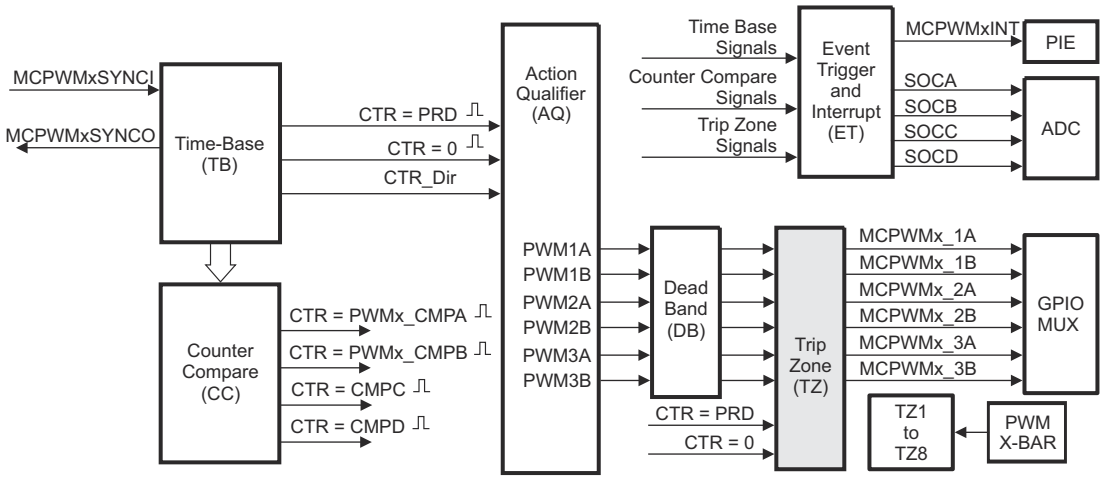


Figure 15-32. Trip-Zone Submodule

15.8.1 Purpose of the Trip-Zone Submodule

The key functions of the trip-zone submodule are:

- Trip inputs $\overline{TZ1}$ to $\overline{TZ8}$ can be flexibly mapped to any MCPWM module.
- Upon a fault condition, outputs MCPWMx_yA and MCPWMx_yB can be forced to one of the following:
  - High
  - Low
  - High-impedance
  - No action taken
- Support for one-shot trip (OST) for major short circuits or over-current conditions.
- Support for cycle-by-cycle tripping (CBC) for current limiting operation.
- Each trip-zone input can be allocated to either one-shot or cycle-by-cycle operation.
- Interrupt generation is possible on any trip-zone input.
- The trip-zone submodule can be fully bypassed if the trip-zone submodule is not required.

15.8.2 Operational Highlights for the Trip-Zone Submodule

The following sections describe the operational highlights and configuration options for the trip-zone submodule.

The trip-zone signals $\overline{TZ1}$ to $\overline{TZ8}$ (also collectively referred to as $\overline{TZn}$) are active-low input signals. When one of these signals goes low, the indication is that a trip event has occurred. Each MCPWM module can be individually configured to ignore or use each of the trip-zone signals. Note that the trip zone settings for all PWM pairs are configured by the same register bits, it is not possible to have separate trip-zone settings for separate PWM pairs within a single MCPWM module. The trip-zone signals used by a particular MCPWM module is configured by the TZSEL register for that specific MCPWM module. The trip-zone signals can be digitally filtered within the GPIO MUX block. A minimum of 3 × TBCLK low pulse width on $\overline{TZn}$ inputs is sufficient to trigger a fault condition on the MCPWM module. If the pulse width is less than this, the trip condition cannot be latched by CBC or OST latches. The asynchronous trip makes sure that if clocks are missing for any reason, the outputs can still be tripped by a valid event present on $\overline{TZn}$ inputs. The GPIOs or peripherals must be appropriately configured. For more information, see the *System Control and Interrupts* chapter.

Each $\overline{TZn}$ input can be individually configured to provide either a cycle-by-cycle or one-shot trip event for an MCPWM module. This configuration is determined by the TZSEL[CBCn], and TZSEL[OSTn] control bits (where n corresponds to the trip input), respectively.

- **Cycle-by-Cycle (CBC):** When a cycle-by-cycle trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the MCPWMx_yA and MCPWMx_yB outputs. The same trip actions are taken on all PWM pairs, denoted by "y". Table 15-11 lists some of the possible actions.

  Additionally, when a cycle-by-cycle trip event occurs, the cycle-by-cycle trip event flag (INTFLAG[CBC]) is set and a PWMx interrupt is generated if the INTEN[CBC] bit is set. A corresponding flag for the specific event that caused the CBC interrupt is also set in the TZCBCOSTFLAG register, which can be cleared by writing to the corresponding flag in TZCBCOSTCLR.

  The specified condition on the inputs is automatically cleared based on the selection made with INTCLR[CBC] if the trip event is no longer present. Therefore, in this mode, the trip event is cleared or reset every PWM cycle. The INTFLG[CBC] and TZCBCOSTFLAG flag bits remain set until the flag bits are manually cleared by writing to the INTCLR[CBC] and TZCBCOSTCLR flag bits. If the cycle-by-cycle trip event is still present when the INTFLG[CBC] and TZCBCOSTFLAG register bits are cleared, then these bits are again immediately set.
- **One-Shot (OST):** When a one-shot trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the MCPWMx_yA and MCPWMx_yB output. The same trip actions are taken on all PWM pairs, denoted by "y". Table 15-11 lists some of the possible actions.

  Additionally, when a one-shot trip event occurs, the one-shot trip event flag (INTFLG[OST]) is set and a PWMx interrupt is generated if the INTEN[OST] bit is set.. A corresponding flag for the event that caused the OST event is also set in register TZCBCOSTFLAG. The one-shot trip condition must be cleared manually by writing to the INTCLR[OST] bit. If desired, the TZCBCOSTFLAG register bit can also be cleared by manually writing to the corresponding bit in the TZCBCOSTCLR register.

Note

Clear the INTFLAG and TZCBCOSTFLAG flags after making sure that the TRIPIN source of the OST has become inactive. Otherwise, if interrupts are enabled, depending on when the flags are cleared, an OST interrupt can occur where the OST flags are zero.

The action taken when a trip event occurs can be configured individually for the A outputs and B outputs of the MCPWM module, but the action is the same across all A outputs and the same across all B outputs. Trip actions are specified by the TZCTL register. Some of the possible actions, shown in Table 15-11, can be taken on a trip event.

Table 15-11. Possible Actions On a Trip Event

TZCTL[TZx] Register Bitfield Settings	MCPWMx_yA and MCPWMx_yB	Comment
0,0	High-Impedance	Tripped
0,1	Force to High State	Tripped
1,0	Force to Low State	Tripped
1,1	No Change	Do Nothing. No change is made to the output.

Example 15-1. Trip-Zone Configurations

Scenario A:
A one-shot trip event on $\overline{TZ1}$ pulls the MCPWM1_yA, MCPWM1_yB outputs low.

- Configure the MCPWM1 registers as follows:
  - TZSEL[OST1] = 1: enables $\overline{TZ1}$ as a one-shot event source for MCPWM1
  - TZCTL[TZA] = 2: MCPWM1_yA is forced low on a trip event.
  - TZCTL[TZB] = 2: MCPWM1_yB is forced low on a trip event.

Scenario B:
A cycle-by-cycle event on $\overline{TZ1}$ or $\overline{TZ6}$ puts MCPWM1_yA into a high-impedance state, MCPWM1_yB is unaffected.

- Configure the MCPWM1 registers as follows:
  - TZSEL[CBC1] = 1: enables $\overline{TZ1}$ as a cycle-by-cycle event source for MCPWM1
  - TZSEL[CBC6] = 1: enables $\overline{TZ6}$ as a cycle-by-cycle event source for MCPWM1
  - TZCTL[TZA] = 0: MCPWM1_yA is put into a high-impedance state on a trip event.
  - TZCTL[TZB] = 3: MCPWM1_yB ignores the trip event.

Note

When configuring the GPIOs and INPUT X-BAR/MCPWM X-BAR options, be aware that a change in the X-BAR input selections can cause an unwanted event. Therefore, set up the GPIO and X-BAR input configurations before enabling the MCPWM Trip-Zone. If a requirement is to change the GPIO/X-BAR configurations while the MCPWM Trip-Zone is enabled, the user can turn off the TRIPs by clearing the TZSEL register and reconfiguring the TRIP selection (TZSEL) after the INPUT X-BAR selection is changed.

15.8.3 Generating Trip Event Interrupts

Figure 15-33 illustrates the trip-zone submodule control logic.

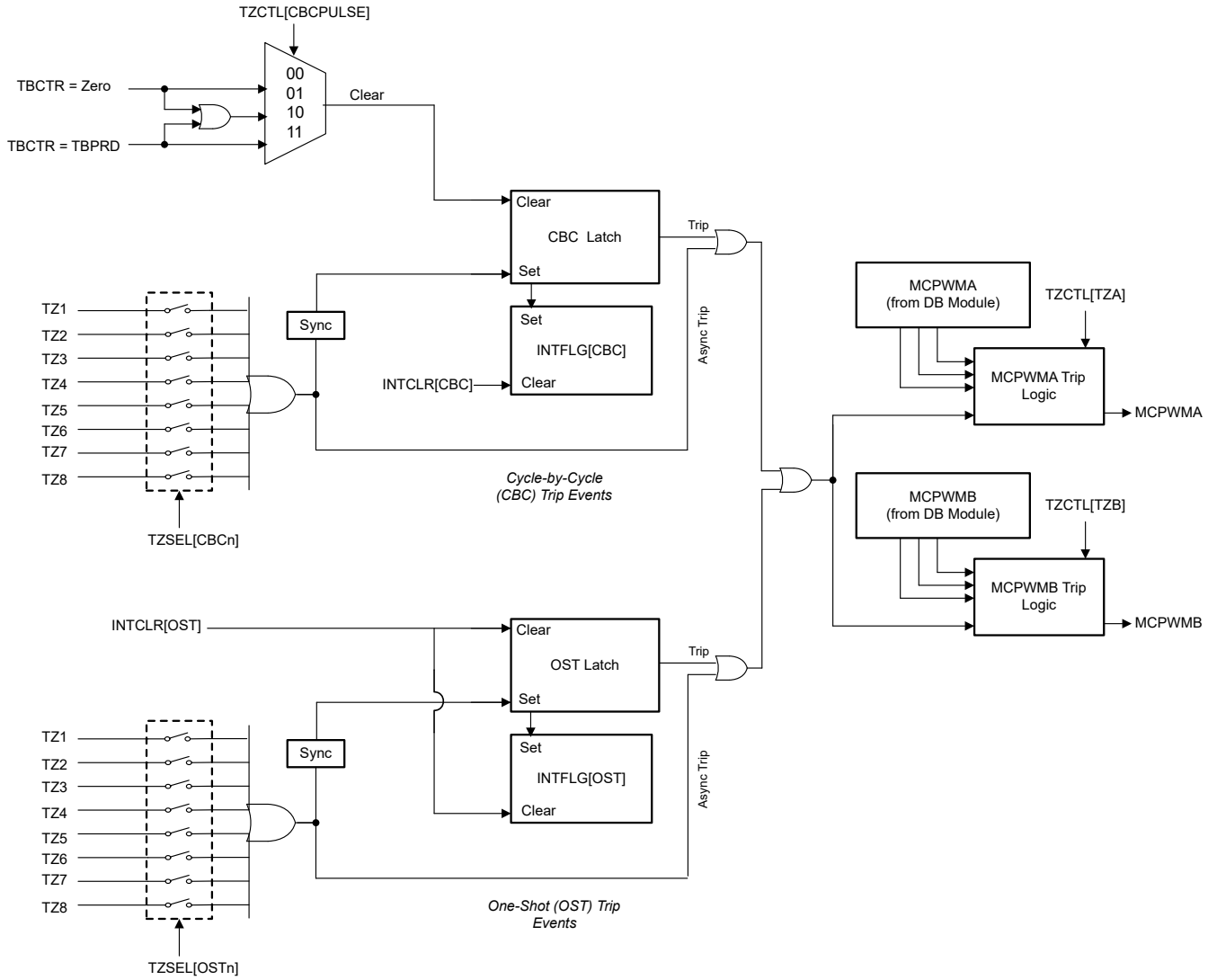


Figure 15-33. Trip-Zone Submodule Mode Control Logic

15.9 Event-Trigger (ET) Submodule

The key functions of the event-trigger submodule are:

- Receives event inputs generated by the time-base and counter-compare submodules
- Uses the time-base direction information for up/down event qualification
- Uses prescaling logic to issue interrupt requests and ADC start of conversion at:
  - Every event
  - Every second event
  - Up to every seventh event
- Uses TZ signals to interrupt the CPU on a trip zone event
- Provides full visibility of event generation using event counters and flags
- Allows software forcing of interrupts

The event-trigger submodule manages the events generated by the time-base submodule, counter-compare submodule, and trip-zone submodule to generate an interrupt to the CPU or a start of conversion pulse to the ADC when a selected event occurs.

Figure 15-34 illustrates the event-trigger submodule within the MCPWM.

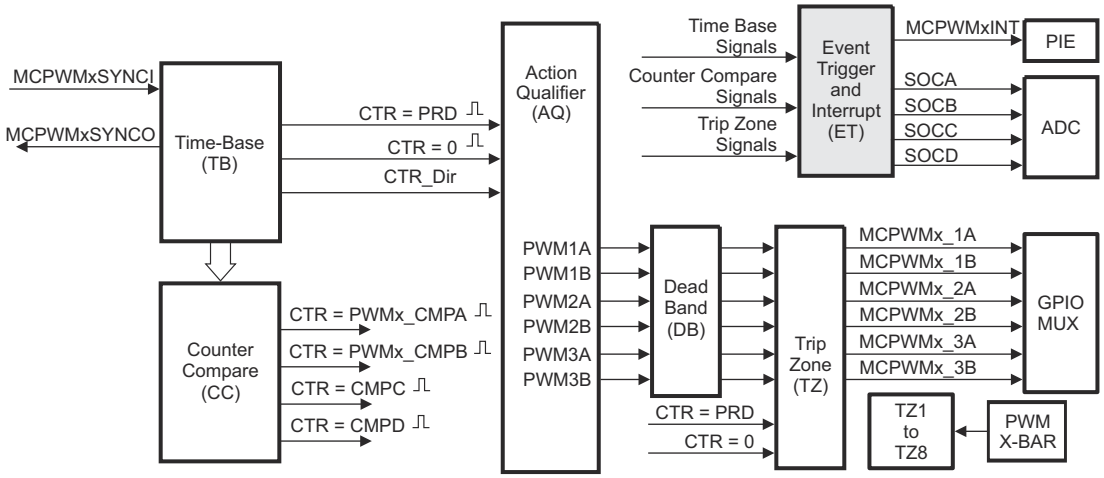


Figure 15-34. Event-Trigger Submodule

15.9.1 Operational Overview of the MCPWM Event-Trigger Submodule

The event-trigger submodule monitors various event conditions (shown as inputs on the left side of Figure 15-35) and can be configured to prescale these events before issuing an Interrupt request or an ADC start of conversion. The event-trigger prescaling logic can issue interrupt requests and ADC start-of-conversion at:

- Every event
- Every second event
- Up to every seventh event

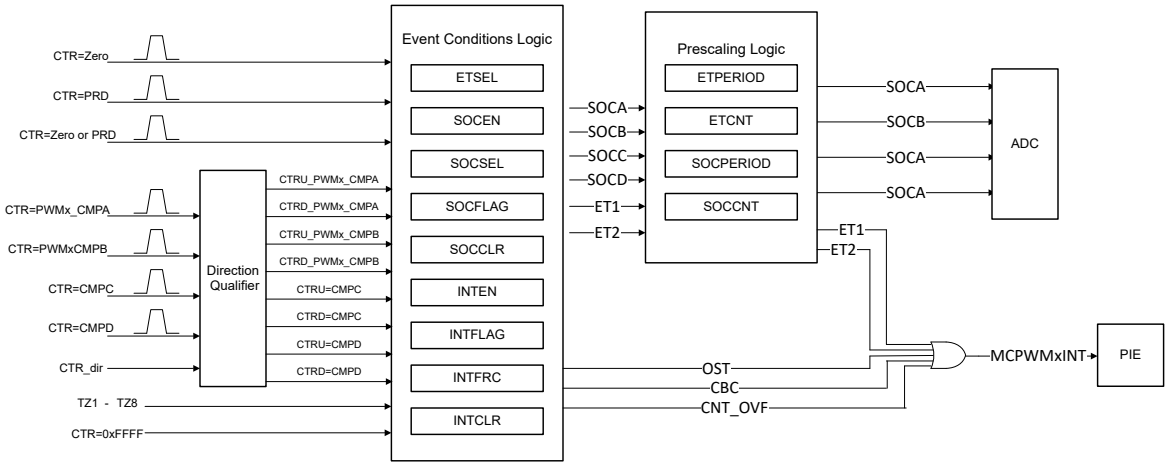


Figure 15-35. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs

- ETSEL - This selects which of the possible events trigger events ET1 or ET2.
- ETPERIOD - This programs the event prescaling options mentioned above.
- ETCNT - This indicates how many events have occurred if the prescaling feature is used.
- INTEN - This enables interrupt sources for the MCPWM.
- INTFLAG - These are flag bits indicating whether an E1x or trip event has occurred.
- INTCLR - These bits allow clearing the flag bits in the INTFLAG register using software.
- INTFRC - These bits allow software forcing of an event. Useful for debugging or software intervention.
- SOCEN - This enables ADC start of conversion events.
- SOCSEL - This selects the trigger sources for the ADC start of conversion events.
- SOCPERIOD - This programs the event prescaling options for SOC events.
- SOCCNT - This indicates how many SOC events have been generated when using the prescaling feature.
- SOCFLAG - This indicates which SOC events have been generated.
- SOCCLR - This clears the SOCFLAG bits when written to using software.

A more detailed look at how the various register bits interact with the interrupt and ADC start-of-conversion logic are shown in Figure 15-36 and Figure 15-37.

Figure 15-36 shows the event-trigger interrupt generation logic. The interrupt-period (ETPERIOD[ETx_PERIOD]) bits specify the number of events required to cause an interrupt pulse to be generated. The choices available are:

- Do not generate an interrupt.
- Generate an interrupt on every event.
- Generate an interrupt on every second event.
- Generate an interrupt on up to every seventh event.

The event that can cause an interrupt is configured by the interrupt enable(INTEN) register and event trigger selection (ETSEL) register. The event can be one of the following:

- Time-base counter equal to zero (TBCTR = 0x00).
- Time-base counter equal to period (TBCTR = TBPRD).
- Time-base counter equal to zero or period (TBCTR = 0x00 || TBCTR = TBPRD).
- Time-base counter equal to the compare A register (CMPA) when the timer is incrementing.
- Time-base counter equal to the compare A register (CMPA) when the timer is decrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is incrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is decrementing.
- Time-base counter equal to the compare C register (CMPC) when the timer is incrementing.
- Time-base counter equal to the compare C register (CMPC) when the timer is decrementing.
- Time-base counter equal to the compare D register (CMPD) when the timer is incrementing.
- Time-base counter equal to the compare D register (CMPD) when the timer is decrementing.
- One-shot trip event.
- Cycle-by-cycle trip event.

The number of events that have occurred can be read from the interrupt event counter ETCNT[ETx_CNT] register bits based off of the selection made using ETSEL. The specified event increments the ETCNT[ETx_CNT] bits until the bits reach the value specified by ETPERIOD[ETx_PERIOD]. When ETPS[INTCNT] = ETPS[INTPRD], the counter stops counting and the counter output is set. The counter is cleared when an INTFLAG[ETx] event is generated.

When ETCNT[ETx_CNT] reaches ETPERIOD[ETx_PERIOD], the following behavior occurs:

If interrupts are enabled, INTEN[ETx] = 1 and the interrupt flag is clear,INTFLAG[ETx] = 0, then an interrupt pulse is generated and the interrupt flag is set, INTFLAG[ETx] = 1. The counter is reset and begins counting events again

- If interrupts are disabled, INTEN[ETx] = 0, or the interrupt flag is set, INTFLAG[ETx] = 1, the counter stops counting events when the counter reaches the period value ETCNT[ETx_CNT] = ETPERIOD[ETx_PERIOD].
- If interrupts are enabled, but the interrupt flag is already set, then the counter holds the output high until the INTFLAG[ETx] flag is cleared. This allows for one interrupt to be pending while one is serviced.

Writing a 0 or a value that is less than the current ETCNT value to the ETPERIOD bits results in the counter going to an undefined state.

The previous definition means that an interrupt on every event up to 7 events if using the ETCNT and ETPERIOD can be generated.

ETINTMIX, ETSOCAMIX and ETSOCBMIX Signals

The event-trigger submodule can generate and use ETINTMIX, ETSOCAMIX, and ETSOCBMIX signals:

- **ETINTMIX:** This signal is a generated from the ORed combination of the sources enabled in the ETINTMIXEN register. The ETINTMIX signal can be used as a source for the PWMx interrupt.
- **ETSOCAMIX:** This signal is a generated from the ORed combination of the sources enabled in the ETSOCAMIXEN register. The ETSOCAMIX signal can be used as a source for the PWMxSOCA trigger signal.
- **ETSOCBMIX:** This signal is a generated from the ORed combination of the sources enabled in the ETSOCBMIXEN register. The ETSOCBMIX signal can be used as a source for the PWMxSOCB trigger signal

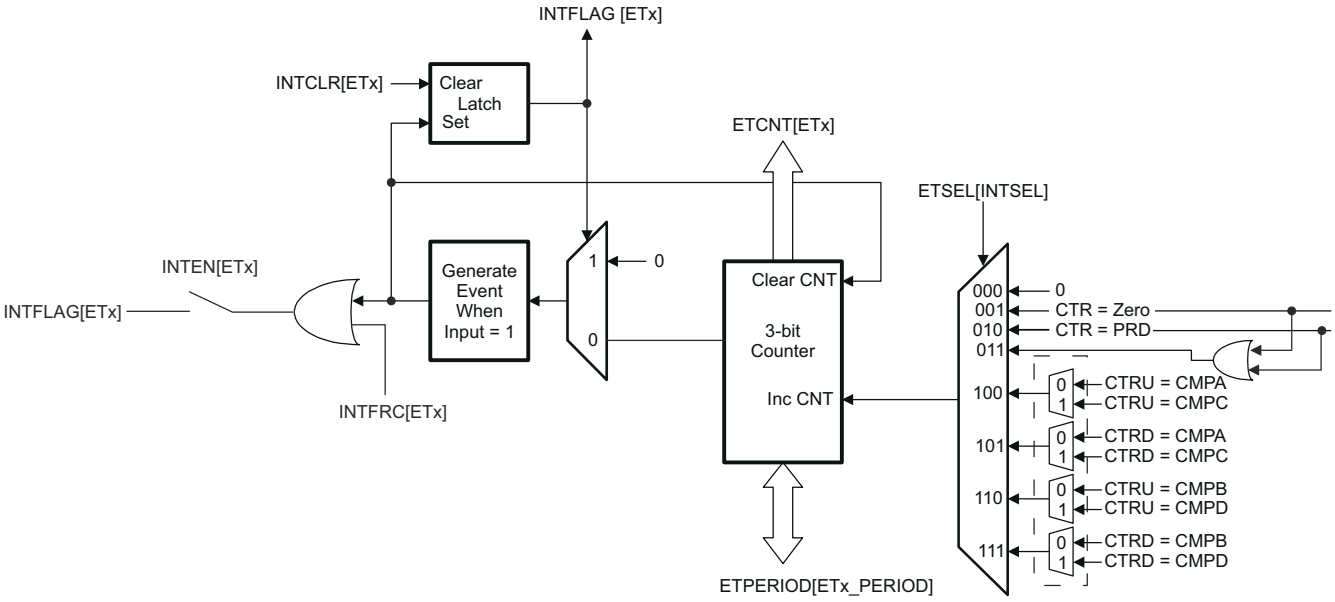


Figure 15-36. Event-Trigger Interrupt Generator

Figure 15-37 shows the operation of the event-trigger's start-of-conversion (SOCx) pulse generator. The SOCCNT[SOCx_CNT] counters and SOCPERIOD[SOCx_PERIOD] period values behave similarly to the interrupt generator except that the pulses are continuously generated. That is, the pulse flag SOCFLAG[SOCx] is latched when a pulse is generated, but the interrupt generator does not stop further pulse generation. The enable and disable bit SOCEN[SOCx_ENABLE] stops pulse generation, but input events can still be counted until the period value is reached as with the interrupt generation logic. The event that triggers an SOCx pulse can be configured separately in the SOCSEL[SOCx_SEL] bits. The possible events are the same events that can be specified for the interrupt generation logic.

Note

SOCC and SOCD events are only available on 6-channel MCPWM. For 2-channel MCPWM, only SOCA and SOCB are available.

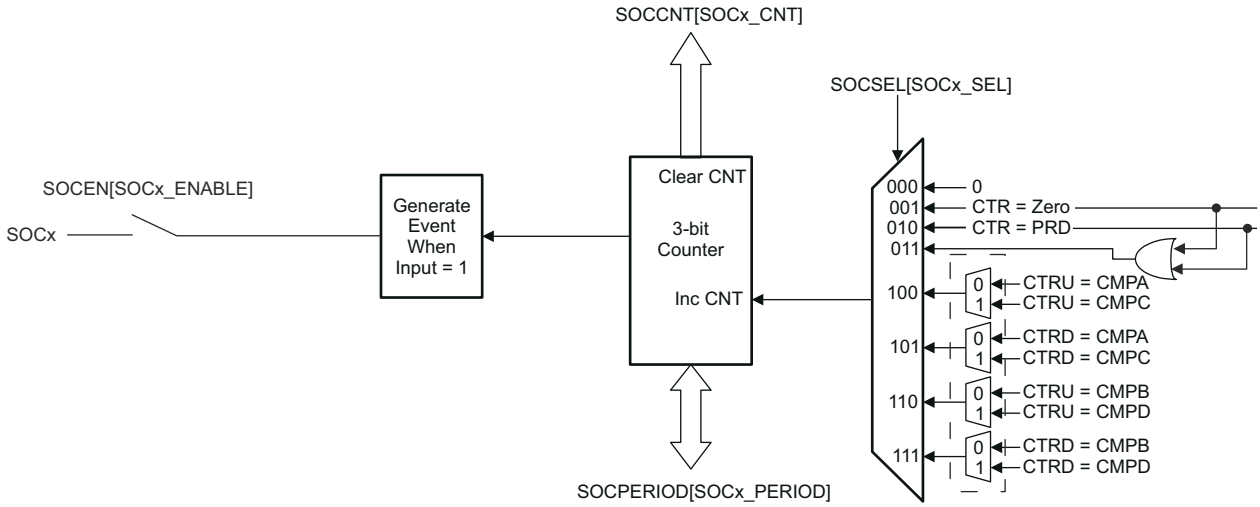


Figure 15-37. Event-Trigger Start-of-Conversion (SOC) Pulse Generator

15.10 PWM Crossbar (X-BAR)

Figure 15-38 shows the architecture of the PWM Crossbar (X-BAR). This module enables selection of various trigger sources into any of the dedicated MCPWM trips inputs.

Note

Refer to the *Crossbar (X-BAR)* chapter for more information on the X-BAR modules, including X-BAR flags.

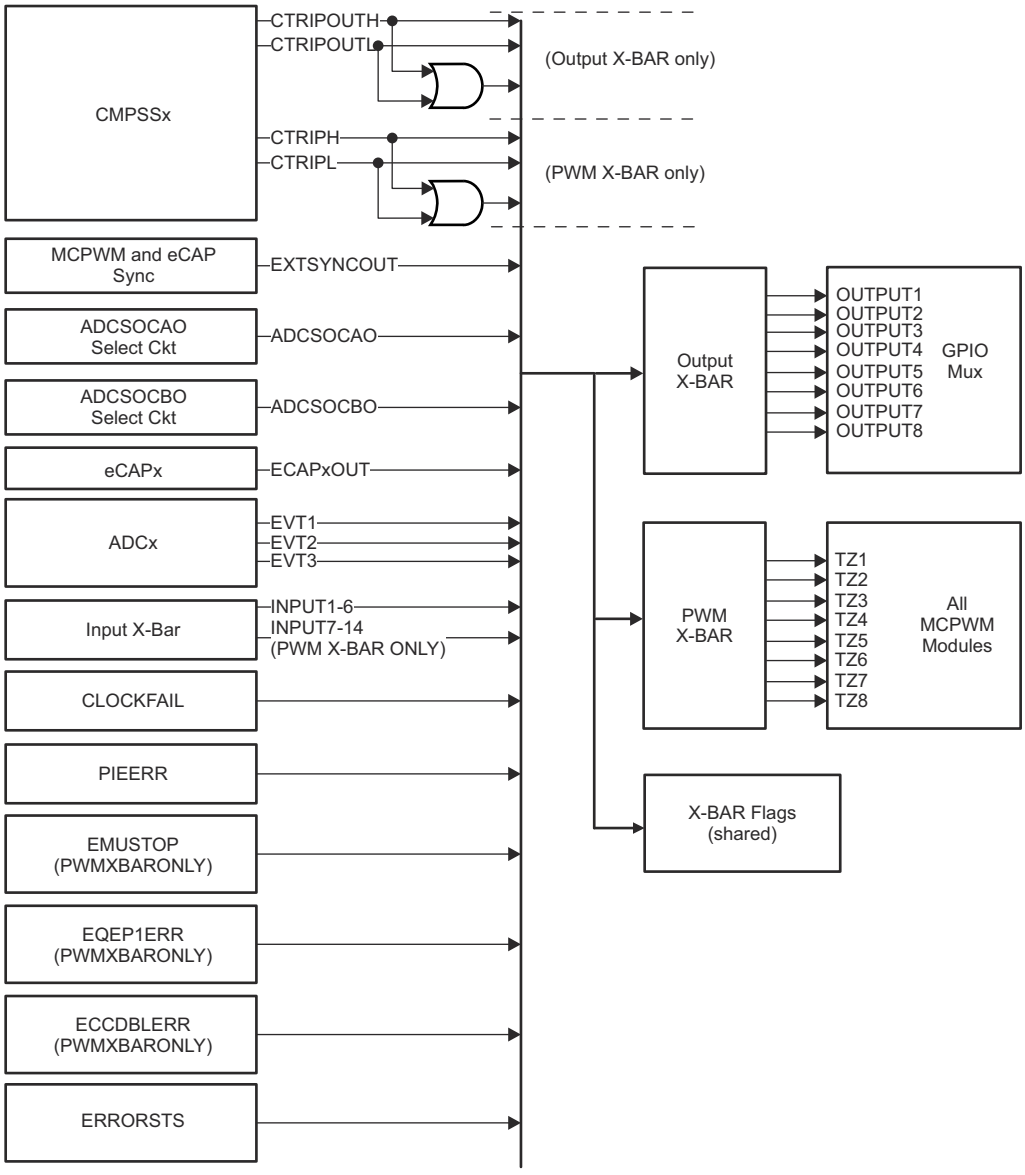


Figure 15-38. PWM X-BAR

The PWM X-BAR has eight outputs that are routed to each MCPWM module. Figure 15-39 represents the architecture of a single output but the output is identical to the architecture of all of the other outputs.

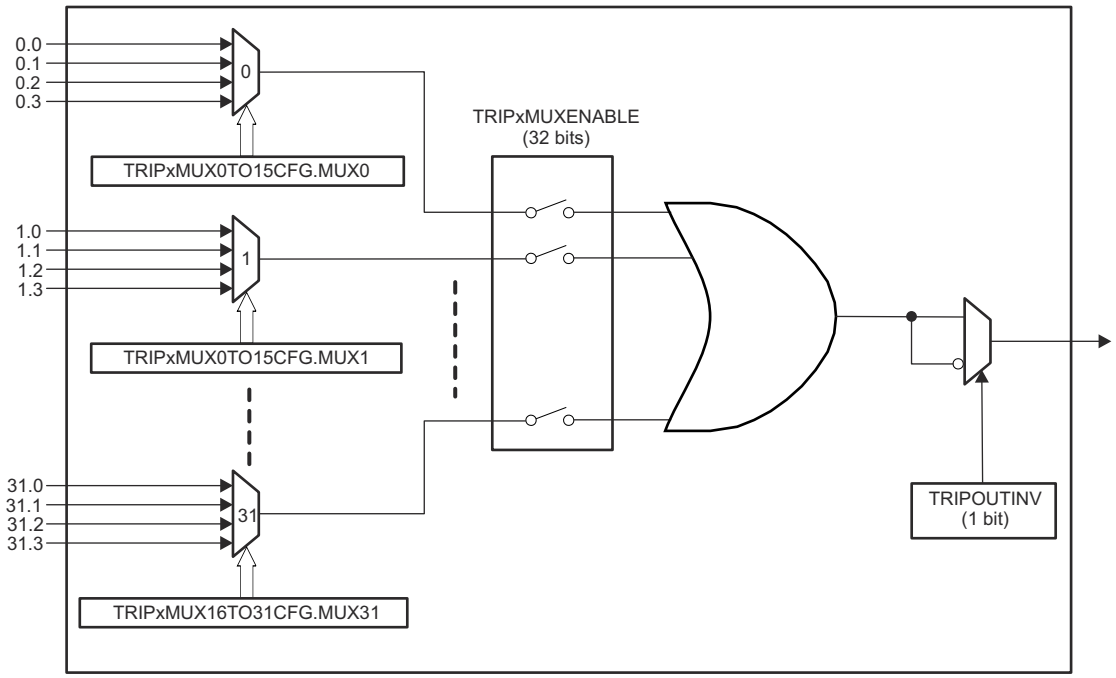


Figure 15-39. PWM X-BAR Architecture - Single Output

First, determine the signals that can be passed to the MCPWM by referencing the PWM X-Bar Mux Configuration Table (see Section 9.2.1.1). Select up to one signal per mux (32 total muxes) for each TRIPx output. Select the inputs to each mux with the TRIPxMUX0TO15CFG and TRIPxMUX16TO31CFG registers. To pass any signal through to the MCPWM, enable the mux in the TRIPxMUXENABLE register. All muxes that are enabled are logically ORed before being passed on to the respective TRIPx signal on the MCPWM. To optionally invert the signal, use the TRIPOUTINV register.

Note

All unused and reserved positions are tied to 0.

15.11 Software

15.11.1 MCPWM Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/mcpwm

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

15.11.1.1 MCPWM Basic PWM Generation and Updates

FILE: mcpwm_ex1_basic_pwm.c

This example configures myMCPWM0 and myMCPWM1 as follows myMCPWM0 Time Base Configurations In Up Count Mode for 10KHz Frequency PWM Action Qualifier is configurations Output A1 : Set High on Counter matches zero, set low on CMPA match Output B1 : Set High on Counter matches zero, set low on CMPB match Output A2 : Set High on Counter matches CMPA, set low on Period match Output B2 : Set High on Counter matches CMPB, set low on Period match Output A3 : Set High on Counter matches CMPA, set low on CMPB match Output B3 : Set Low on Counter matches CMPA, set high on CMPB match Interrupt For Updating the CMPx Values runtime, Interrupt is configured in synchronous with the PWM Counter. ET1 is configured to fire at every 5 events of Counter = Period. ET1 is selected as source for the interrupt generation

myMCPWM1 Time Base Configurations In Up Down Count Mode for 10KHz Frequency PWM Action Qualifier is configurations Output A1 : Set High on Counter matches CMPA while counting up, Set low on CMPA match while Counting down Output B1 : Set High on Counter matches CMPB while counting up, Set low on CMPB match while Counting down Interrupt For Updating the CMPx Values runtime, Interrupt is configured in synchronous with the PWM Counter. ET1 is configured to fire at every 5 events of Counter = Period. ET1 is selected as source for the interrupt generation

External Connections
Monitor the MCPWM pins for the waveforms. Observe J4 40 - GPIO0 - myMCPWM0 Output A1 Observe J4 39 - GPIO1 - myMCPWM0 Output B1 Observe J4 38 - GPIO2 - myMCPWM0 Output A2 Observe J4 37 - GPIO3 - myMCPWM0 Output B2 Observe J4 36 - GPIO4 - myMCPWM0 Output A3 Observe J4 35 - GPIO5 - myMCPWM0 Output B3 Observe J4 7 - GPIO12 - myMCPWM1 Output A1 Observe J2 19 - GPIO29 - myMCPWM1 Output B1

15.11.1.2 MCPWM Basic PWM Generation and Updates

FILE: mcpwm_ex2_synchronization.c

This example configures myMCPWM0 and myMCPWM1 as follows myMCPWM0 Time Base Configurations In Up Count Mode for 10KHz Frequency PWM or 15999 TBCLKs for TBCLK of 160MHz Enabled Sync in for the InputXbar 5, with phase shift load enabled Sync Out is configured for every Counter equals Period event Action Qualifier is configurations Output A1 : Set High on Counter matches zero, set low on CMPA match Output B1 : Set High on Counter matches zero, set low on CMPB match Interrupt For Updating the Phase Shift Values runtime, Interrupt is configured in synchronous with the PWM Counter. ET2 is configured to fire at every 5 events of Counter = Period. ET2 is selected as source for the interrupt generation

myMCPWM1 Time Base Configurations In Up Down Count Mode for 10KHz Frequency PWM or 8000 TBCLKs for TBCLK of 160MHz Enabled Sync in from myMCPWM0, with phase shift load enabled. Default Counter Mode after sync is set to counting up. But can be changed. Action Qualifier is configurations Output A1 : Set High on Counter matches zero, set low on CMPA match Output B1 : Set High on Counter matches zero, set low on CMPB match

External Connections
Monitor the MCPWM pins for the waveforms. Observe J4 40 - GPIO0 - myMCPWM0 Output A1 Observe J4 39 - GPIO1 - myMCPWM0 Output B1 Observe J4 7 - GPIO12 - myMCPWM1 Output A1 Observe J2 19 - GPIO29 - myMCPWM1 Output B1

Watch Variables

- gUsePhaseValueUpdate - set to true to runtime update value for the phase shift value.
- gPhaseShiftValue - value to be used for the phase shift value.
- gPhaseDirUp - true to use up phase shift value, false for down phase shift value.

15.11.1.3 MCPWM Basic PWM generation With DeadBand

FILE: mcpwm_ex3_deadband.c

This example configures myMCPWM0 and myMCPWM1 as follows myMCPWM0 Time Base Configurations In Up Down Count Mode for 4000 TBCLKs or 20KHz Frequency for TBCLK of 160MHz Action Qualifier is configurations Output A1 : Set High on Counter matches CMPA counting up, set low on CMPA match counting down Output A2 : Set High on Counter matches CMPA counting up, set low on CMPA match counting down Output A3 : Set High on Counter matches CMPA counting up, set low on CMPA match counting down DeadBand Active High Configuration is selected with 100 TBCLK deadband

myMCPWM1 Time Base Configurations In Up Down Count Mode for 4000 TBCLKs or 20KHz Frequency for TBCLK of 160MHz Action Qualifier is configurations Output A1 : Set High on Counter matches CMPA counting up, set low on CMPA match counting down DeadBand Disabled Configurations

External Connections
Monitor the MCPWM pins for the waveforms. Observe J4 40 - GPIO0 - myMCPWM0 Output A1 - Active High Configuration Observe J4 39 - GPIO1 - myMCPWM0 Output B1 - Active High Configuration Observe J4 38 - GPIO2 - myMCPWM0 Output A2 - Active High Configuration Observe J4 37 - GPIO3 - myMCPWM0 Output B2 - Active High Configuration Observe J4 36 - GPIO4 - myMCPWM0 Output A3 - Active High Configuration Observe J4 35 - GPIO5 - myMCPWM0 Output B3 - Active High Configuration Observe J4 7 - GPIO12 - myMCPWM1 Output A1 - Can be used for reference

15.11.1.4 MCPWM Basic PWM Generation and Updates without Sysconfig

FILE: mcpwm_ex4_basic_pwm_no_syscfg.c

This example configures myMCPWM0 and myMCPWM1 as follows myMCPWM0 Time Base Configurations In Up Count Mode for 10KHz Frequency PWM Action Qualifier is configurations Output A1 : Set High on Counter matches zero, set low on CMPA match Output B1 : Set High on Counter matches zero, set low on CMPB match Output A2 : Set High on Counter matches CMPA, set low on Period match Output B2 : Set High on Counter matches CMPB, set low on Period match Output A3 : Set High on Counter matches CMPA, set low on CMPB match Output B3 : Set Low on Counter matches CMPA, set high on CMPB match Interrupt For Updating the CMPx Values runtime, Interrupt is configured in synchronous with the PWM Counter. ET1 is configured to fire at every 5 events of Counter = Period. ET1 is selected as source for the interrupt generation

myMCPWM1 Time Base Configurations In Up Down Count Mode for 10KHz Frequency PWM Action Qualifier is configurations Output A1 : Set High on Counter matches CMPA while counting up, Set low on CMPA match while Counting down Output B1 : Set High on Counter matches CMPB while counting up, Set low on CMPB match while Counting down Interrupt For Updating the CMPx Values runtime, Interrupt is configured in synchronous with the PWM Counter. ET1 is configured to fire at every 5 events of Counter = Period. ET1 is selected as source for the interrupt generation

External Connections
Monitor the MCPWM pins for the waveforms. Observe J4 40 - GPIO0 - myMCPWM0 Output A1 Observe J4 39 - GPIO1 - myMCPWM0 Output B1 Observe J4 38 - GPIO2 - myMCPWM0 Output A2 Observe J4 37 - GPIO3 - myMCPWM0 Output B2 Observe J4 36 - GPIO4 - myMCPWM0 Output A3 Observe J4 35 - GPIO5 - myMCPWM0 Output B3 Observe J4 7 - GPIO12 - myMCPWM1 Output A1 Observe J2 19 - GPIO29 - myMCPWM1 Output B1

15.11.1.5 MCPWM PWM Tripzone Feature Showcase

FILE: mcpwm_ex5_tripzone.c

This example configures myMCPWM0 and myMCPWM1 as follows myMCPWM0 Time Base Configurations In Up Count Mode for 15999 TBCLKs or 10KHz Frequency for TBCLK of 160MHz Action Qualifier configurations

Output A1/B1: Set High on Counter Zero, Set Low on Counter matches CMPA/CMPB counting up Output A2/B2: Set Low on Counter Period, Set High on Counter matches CMPA/CMPB counting up Output A3: Set High on Counter matches CMPA, Set Low on Counter matches CMPB counting up Output B3: Set Low on Counter matches CMPA, Set High on Counter matches CMPB counting up Trip Zone Configurations CBC1 is source of the CBC trip. CBC Clear is "counter = zero" event TZA is configured for Setting Low, TZB is configured to Setting High Interrupt Configurations Interrupt is configured for CBC trip

myMCPWM1 Time Base Configurations In Up Down Count Mode for 8000 TBCLKs or 10KHz Frequency for TBCLK of 160MHz Action Qualifier configurations Output A1: Set High on Counter matches CMPA counting up, Set Low on Counter matches CMPA counting down Output B1: Set High on Counter matches CMPB counting up, Set Low on Counter matches CMPB counting down Trip Zone Configurations OSHT1 is source of the OneShot trip TZA is configured for Setting Low, TZB is configured to Setting High Interrupt Configurations Interrupt is configured for OST trip

myGPIO0 and InputXbar and PwmXbar GPIO6 is configured in Output Mode and is connected to InputXbar13 This InputXbar instance is connected to the PwmXbar instance that connects to trip This is used for triggering the Trip Input Trip Signal Routing +-----+ +--------------+ +----+ +------------+ | | | | | | -->| CBC1 (MCPWM0) | | GPIO6 |--->| INPUTXBAR_IN13 |--->|PWMXBAR|| +------------+ | | | | | | -->| OSHT1(MCPWM1) | +-----+ +--------------+ +----+ +------------+

If external trip is used by the user, please configure GPIO6 as input mode and remove the GPIO6 driving code from the example *External Connections*
Observe the trip input on GPIO6: Observe J4 32 - GPIO6 - Trip Input Monitor the MCPWM pins for the waveforms: Observe J4 40 - GPIO0 - myMCPWM0 Output A1 Observe J4 39 - GPIO1 - myMCPWM0 Output B1 Observe J4 38 - GPIO2 - myMCPWM0 Output A2 Observe J4 37 - GPIO3 - myMCPWM0 Output B2 Observe J4 36 - GPIO4 - myMCPWM0 Output A3 Observe J4 35 - GPIO5 - myMCPWM0 Output B3 Observe J1 7 - GPIO12 - myMCPWM1 Output A1 Observe J2 19 - GPIO29 - myMCPWM1 Output B1

15.11.1.6 MCPWM Global Load Feature Showcase

FILE: mcpwm_ex6_global_load_use_case.c

This example configures myMCPWM0 and myMCPWM1 with global load feature for synchronized parameter updates across multiple PWM channels as follows: myMCPWM0 (Primary Module) Time Base Configurations: - Up Count Mode - Period = 3999 (40 kHz @ TBCLK of 160MHz) - Free Run emulation mode Initial Compare Values: - PWM1A/B : 1800 (45% duty cycle) - PWM2A/B : 3133 (78% duty cycle) - PWM3A/B : 467 (12% duty cycle) Action Qualifier Configurations: All pairs (A/B outputs): - Set HIGH on Counter matches CMPA/CMPB counting up - Set LOW on Counter Period Deadband Configurations: - Initial: 32 counts (200ns @ 160MHz) - Both Rising and Falling edge delay enabled myMCPWM1 (Secondary Module) Time Base Configurations: - Up Count Mode synchronized to myMCPWM0 - Period = 3999 (40 kHz) Initial Compare Values: - PWM1A/B : 1000 (25% duty cycle) Action Qualifier Configurations: PWM1A/B outputs: - Set HIGH on Counter matches CMPA/CMPB counting up - Set LOW on Counter Period Global Load Feature Implementation: When perform_one_shot_load flag is set: 1. Compare Value Updates: myMCPWM0: - PWM1A: 2000 (50% duty) - PWM2A: 3333 (83% duty) - PWM3A: 667 (17% duty) myMCPWM1: - PWM1A: 2000 (50% duty) 2. Deadband Updates (both modules): - 48 counts (300ns @ 160MHz) - Applied to both rising and falling edges 3. Update Mechanism: - All changes written to shadow registers - Global load trigger synchronizes all updates - Changes take effect simultaneously

External Connections
Monitor the MCPWM pins for the waveforms: Observe J4 40 - GPIO0 - myMCPWM0 Output A1 Observe J4 39 - GPIO1 - myMCPWM0 Output B1 Observe J4 38 - GPIO2 - myMCPWM0 Output A2 Observe J4 37 - GPIO3 - myMCPWM0 Output B2 Observe J4 36 - GPIO4 - myMCPWM0 Output A3 Observe J4 35 - GPIO5 - myMCPWM0 Output B3 Observe J1 7 - GPIO12 - myMCPWM1 Output A1 Observe J2 19 - GPIO29 - myMCPWM1 Output B1

- Use an oscilloscope to observe synchronized updates
- Watch deadband transitions on complementary pairs

- All updates occur on the next global load event
- Period remains constant during updates

15.11.1.7 MCPWM DMA Configuration for Dynamic PWM Control

FILE: mcpwm_ex7_dma.c

This example demonstrates how to configure MCPWM modules and DMA controllers to implement dynamic updates of PWM parameters without CPU intervention:

MCPWM Configuration:
- MCPWM0 and MCPWM1 are configured for PWM output generation
- MCPWM0 provides SOC trigger for DMA CH1 operations
- MCPWM1 provides SOC trigger for DMA CH2 operations

DMA Configuration:
- DMA CH1 is configured to update MCPWM0 compare registers (PWM1-3 CMPAs) in burst mode, transferring 3 words per trigger
- DMA CH2 is configured to update MCPWM1 compare registers (PWM1 CMPA) in burst mode, transferring 1 word per trigger
- Both DMA channels operate continuously, cycling through 4 different configurations stored in memory
- The example demonstrates advanced PWM pattern generation without CPU overhead by allowing DMA to dynamically update multiple PWM parameters on each PWM period

This technique is particularly useful for:
- Generating complex PWM patterns for motor control
- Creating arbitrary waveforms with precise timing
- Implementing efficient, real-time PWM updates for dynamic system control

External Connections
Monitor the MCPWM pins for the waveforms. Observe J4 40 - GPIO0 - myMCPWM0 Output A1 Observe J4 39 - GPIO1 - myMCPWM0 Output B1 Observe J4 38 - GPIO2 - myMCPWM0 Output A2 Observe J4 37 - GPIO3 - myMCPWM0 Output B2 Observe J4 36 - GPIO4 - myMCPWM0 Output A3 Observe J4 35 - GPIO5 - myMCPWM0 Output B3 Observe J4 7 - GPIO12 - myMCPWM1 Output A1 Observe J2 19 - GPIO29 - myMCPWM1 Output B1

Watch Variables
- compareConfigs_6ch[] - Array containing MCPWM0 compare values for PWM1-3 CMPAs
- compareConfigs_2ch[] - Array containing MCPWM1 compare values for PWM1 CMPA

15.12 MCPWM Registers

This Section describes the MCPWM Registers.

15.12.1 MCPWM Base Address Table

Table 15-12. MCPWM Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
Pwm1Regs	MCPWM_6CH_REGS	PWM1_BASE	0x0000_4000	YES	YES
Pwm3Regs	MCPWM_2CH_REGS	PWM3_BASE	0x0000_4800	YES	YES

15.12.2 MCPWM_6CH_REGS Registers

Table 15-13 lists the memory-mapped registers for the MCPWM_6CH_REGS registers. All register offset addresses not listed in Table 15-13 should be considered as reserved locations and the register contents should not be modified.

Table 15-13. MCPWM_6CH_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	REVISION	IP revision id register	
8h	TBCTL	Time base control register	
Ah	TBPRD	Time base period register	
Ch	TBPRDS	Time base period shadow register	
Eh	TBPHS	Time base phase offset register	
10h	TBSTS	Time base status register	
12h	TBSTSCLR	Time base status clear register	
14h	TBCTR	Time base counter register	
18h	CMPCTL	Counter compare control register	
20h	CMPC	Counter compare C register	
22h	CMPD	Counter compare D register	
24h	CMPCS	Counter compare C shadow register	
26h	CMPDS	Counter compare D shadow register	
28h	AQCTL	Action qualifier control register	
30h	SOCEN	Start of conversion enable	
32h	SOCSEL	Start of conversion selection	
34h	SOCPERIOD	Start of conversion period	
36h	SOCCNT	Start of conversion count	
38h	SOCFLAG	Start of conversion flag	
3Ah	SOCCLR	Start of conversion clear	
40h	ETSEL	Event trigger selection	
42h	ETPERIOD	Event trigger period	
44h	ETCNT	Event trigger count	
48h	INTEN	Interrupt Enable Register	EALLOW
4Ah	INTFLAG	Interrupt Flag Register	
4Ch	INTCLR	Interrupt Clear Register	EALLOW
4Eh	INTFRC	Interrupt Force Register	EALLOW
50h	TZSEL	Trip Zone Selection	EALLOW
56h	TZCTL	Trip Zone control	EALLOW
58h	TZCBCOSTFLAG	Trip Zone CBC and OST Flag Register	
5Ah	TZCBCOSTCLR	Trip Zone CBC and OST Clear Register	EALLOW
60h	DBCTL	Dead band control register	
68h	DBFED	Dead band fall edge delay	
6Ah	DBRED	Dead band rise edge delay	
6Ch	DBFEDS	Dead band fall edge delay shadow register	
6Eh	DBREDS	Dead band rise edge delay shadow register	
78h	GLDCTL	Global load control register	EALLOW
7Ah	GLDOSHTCTL	Global load one shot control register	
7Ch	GLDOSHTSTS	Global load one shot status register	
80h	PWM1_CMPA	PWM1 counter compare A register	

Table 15-13. MCPWM_6CH_REGS Registers (continued)

Offset	Acronym	Register Name	Write Protection
82h	PWM1_CMPAS	PWM1 counter compare A shadow register	
84h	PWM1_CMPB	PWM1 counter compare B register	
86h	PWM1_CMPBS	PWM1 counter compare B shadow register	
90h	PWM1_AQCTLA	PWM1 action qualifier A register	
92h	PWM1_AQCTLAS	PWM1 action qualifier A shadow register	
94h	PWM1_AQCTLB	PWM1 action qualifier B register	
96h	PWM1_AQCTLBS	PWM1 action qualifier B shadow register	
98h	PWM1_AQSFRC	PWM1 action qualifier software force	
9Ah	PWM1_AQOTSFRC	PWM1 action qualifier one time software force	
180h	PWM2_CMPA	PWM2 counter compare A register	
182h	PWM2_CMPAS	PWM2 counter compare A shadow register	
184h	PWM2_CMPB	PWM2 counter compare B register	
186h	PWM2_CMPBS	PWM2 counter compare B shadow register	
190h	PWM2_AQCTLA	PWM2 action qualifier A register	
192h	PWM2_AQCTLAS	PWM2 action qualifier A shadow register	
194h	PWM2_AQCTLB	PWM2 action qualifier B register	
196h	PWM2_AQCTLBS	PWM2 action qualifier B shadow register	
198h	PWM2_AQSFRC	PWM2 action qualifier software force	
19Ah	PWM2_AQOTSFRC	PWM2 action qualifier one time software force	
280h	PWM3_CMPA	PWM3 counter compare A register	
282h	PWM3_CMPAS	PWM3 counter compare A shadow register	
284h	PWM3_CMPB	PWM3 counter compare B register	
286h	PWM3_CMPBS	PWM3 counter compare B shadow register	
290h	PWM3_AQCTLA	PWM3 action qualifier A register	
292h	PWM3_AQCTLAS	PWM3 action qualifier A shadow register	
294h	PWM3_AQCTLB	PWM3 action qualifier B register	
296h	PWM3_AQCTLBS	PWM3 action qualifier B shadow register	
298h	PWM3_AQSFRC	PWM3 action qualifier software force	
29Ah	PWM3_AQOTSFRC	PWM3 action qualifier one time software force	

Complex bit access types are encoded to fit into small table cells. Table 15-14 shows the codes that are used for access types in this section.

Table 15-14. MCPWM_6CH_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 15-14. MCPWM_6CH_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

15.12.2.1 REVISION Register (Offset = 0h) [Reset = 00000007h]

REVISION is shown in Figure 15-40 and described in Table 15-15.

Return to the Summary Table.

IP revision id register

Figure 15-40. REVISION Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				CAP_PRESENT	CMPCD_PRESENT	PWM3_PRESENT	PWM2_PRESENT
R-0-0h				R-0h	R-1h	R-1h	R-1h

Table 15-15. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	CAP_PRESENT	R	0h	This hardcoded field defines the presence of Capture mode feature. Reset type: SYSRSn
2	CMPCD_PRESENT	R	1h	This hardcoded field defines the presence of Compare C and D registers. Reset type: SYSRSn
1	PWM3_PRESENT	R	1h	This hardcoded field defines the presence of PWM3 channels. Reset type: SYSRSn
0	PWM2_PRESENT	R	1h	This hardcoded field defines the presence of PWM2 channels. Reset type: SYSRSn

15.12.2.2 TBCTL Register (Offset = 8h) [Reset = 00000002h]

TBCTL is shown in Figure 15-41 and described in Table 15-16.

Return to the Summary Table.

Time base control register

Figure 15-41. TBCTL Register

31	30	29	28	27	26	25	24
RESERVED		SYNCISEL					SYNCPERSEL
R-0-0h		R/W-0h					R/W-0h
23	22	21	20	19	18	17	16
SYNCPERSEL		FREE_SOFT		RESERVED	SYNCOSEL		
R/W-0h		R/W-0h		R-0-0h	R/W-0h		
15	14	13	12	11	10	9	8
SWSYNC	RESERVED			PHSDIR	PHSEN	RESERVED	PRDLD
R-0/W1S-0h	R-0-0h			R/W-0h	R/W-0h	R-0-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		CLKDIV				CTRMODE	
R-0-0h		R/W-0h				R/W-2h	

Table 15-16. TBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	Reserved
29-25	SYNCISEL	R/W	0h	These bits determines the source of SYNCIN signal. 0x0 : Disabled using SOC tieoff. 0x1-0x1F : Refer to PWM chapter of TRM. Reset type: SYSRSn
24-22	SYNCPERSEL	R/W	0h	Sync peripheral Select 000: Reserved (Disabled) 001: Reserved (Disabled) 010: CTR = PRD 011: CTR = 0 100: CTR = CMPC, Count direction Up 101: CTR = CMPC, Count direction Down 110: CTR = CMPD, Count direction Up 111: CTR = CMPD, Count direction Down Reset type: SYSRSn
21-20	FREE_SOFT	R/W	0h	Emulation Mode Bits. These bits select the behavior of the time-base counter during emulation events 00: Stop after the next time-base counter increment or decrement 01: Stop when counter completes a whole cycle: - Up-count mode: stop when the time-base counter = period (TBCTR = TBPRD) - Up-down-count mode: stop when the time-base counter = 0x00 (TBCTR = 0x00) 1x: Free run Reset type: SYSRSn
19	RESERVED	R-0	0h	Reserved

Table 15-16. TBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-16	SYNCOSEL	R/W	0h	Sync Output Select 000: SWFSYNC 001: CTR = zero: Time-base counter equal to zero (TBCTR = 0x00) 010: CTR = CMPC : Time-base counter equal to counter-compare C (TBCTR = CMPC) 011: CTR = CMPD : Time-base counter equal to counter-compare D (TBCTR = CMPD) 1xx: Disabled MCPWMxSYNCO sync signal Reset type: SYSRSn
15	SWSYNC	R-0/W1S	0h	Software Forced Sync Pulse 0: Writing a 0 has no effect and reads always return a 0. 1: Writing a 1 forces a one-time synchronization pulse to be generated. Reset type: SYSRSn
14-12	RESERVED	R-0	0h	Reserved
11	PHSDIR	R/W	0h	Phase Direction Bit. This bit is only used when the time-base counter is configured in the up-down-count mode. In the up-count mode, this bit is ignored. The bit indicates the direction of the time-base counter (TBCTR) after a sync event occurs and a new phase value is loaded from the phase (TBPHS) register. 0: Count down after the sync event. 1: Count up after the sync event. Reset type: SYSRSn
10	PHSEN	R/W	0h	Load Phase register to time-base counter(TBCNTR) 0: Do not load the time-base counter (TBCTR) from the time-base phase register (TBPHS). 1: Allow Counter to be loaded from the Phase register (TBPHS) when an MCPWMxSYNCI input signal occurs or a software-forced sync (SWSYNC). Reset type: SYSRSn
9	RESERVED	R-0	0h	Reserved
8	PRDLD	R/W	0h	Shadow to Active load of TBPRD register 0: Shadow to Active Load of TBPRD occurs when TBCTR = 0 1 : Disabled shadow to active load of TBPRD Reset type: SYSRSn
7-6	RESERVED	R-0	0h	Reserved
5-2	CLKDIV	R/W	0h	Time Base Clock Pre-Scale Bits These bits select the time base clock pre-scale value (TBCLK = MCPWMCLK/CLKDIV): 0000: /1 (default on reset) 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 1001: /512 1010: /1024 1011: /2048 1100: /4096 1101: /8192 1110: /16384 1111: /32768 Reset type: SYSRSn

Table 15-16. TBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	CTRMODE	R/W	2h	The time-base counter mode is normally configured once and not changed during normal operation. If you change the mode of the counter, the change will take effect at the next TBCLK edge and the current counter value shall increment or decrement from the value before the mode change. These bits set the time-base counter mode of operation as follows: 00: Up-count mode 01: Up-down count mode 1x: Freeze counter operation (default on reset) Reset type: SYSRSn

15.12.2.3 TBPRD Register (Offset = Ah) [Reset = 00000000h]

TBPRD is shown in Figure 15-42 and described in Table 15-17.

Return to the Summary Table.

Time base period register

Figure 15-42. TBPRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TBPRD															
R-0-0h																R/W-0h															

Table 15-17. TBPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBPRD	R/W	0h	Time Base Period Register These bits determine the period of the time-base counter. This sets the PWM frequency. Shadowing of this register is enabled and disabled by the TBCTL[PRDLD] bit. By default this register is shadowed. - If TBCTL[PRDLD] = 0, then the shadow is enabled. This register will also loaded from the shadow register when the time-base counter equals zero. - If TBCTL[PRDLD] = 1, then the shadow is disabled. Reset type: SYSRSn

15.12.2.4 TBPRDS Register (Offset = Ch) [Reset = 00000000h]

TBPRDS is shown in Figure 15-43 and described in Table 15-18.

Return to the Summary Table.

Time base period shadow register

Figure 15-43. TBPRDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TBPRDS															
R-0-0h																R/W-0h															

Table 15-18. TBPRDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBPRDS	R/W	0h	Time Base Period Shadow Register The value in the TBPRDS register is loaded into TBPRD register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.5 TBPHS Register (Offset = Eh) [Reset = 00000000h]

TBPHS is shown in Figure 15-44 and described in Table 15-19.

Return to the Summary Table.

Time base phase offset register

Figure 15-44. TBPHS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TBPHS															
R-0-0h																R/W-0h															

Table 15-19. TBPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBPHS	R/W	0h	Phase Offset Register These bits set time-base counter phase of the PWM relative to the sync (MCPWMxSYNCI / SWFSYNC) - If TBCTL[PHSEN] = 0, then the sync event is ignored. - If TBCTL[PHSEN] = 1, then the time-base counter (TBCTR) will be loaded with the phase (TBPHS) when a sync event occurs. The sync event can be initiated by the input sync signal (MCPWMxSYNCI) or by a software forced sync. Reset type: SYSRSn

15.12.2.6 TBSTS Register (Offset = 10h) [Reset = 00000000h]

TBSTS is shown in Figure 15-45 and described in Table 15-20.

Return to the Summary Table.

Time base status register

Figure 15-45. TBSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						SYNCI	CTRDIR
R-0-0h						R-0h	R-0h

Table 15-20. TBSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	SYNCI	R	0h	Input Synchronization Latched Status Bit 0: No external sync event has occurred. 1: External sync event has occurred (MCPWMxSYNCI). Reset type: SYSRSn
0	CTRDIR	R	0h	Time Base Counter Direction Status Bit 0: Time-Base Counter is currently counting up. 1: Time-Base Counter is currently counting down. Note: This bit is only valid when the counter is not frozen. Reset type: SYSRSn

15.12.2.7 TBSTSCLR Register (Offset = 12h) [Reset = 00000000h]

TBSTSCLR is shown in Figure 15-46 and described in Table 15-21.

Return to the Summary Table.

Time base status clear register

Figure 15-46. TBSTSCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						SYNCI	RESERVED
R-0-0h						R-0/W1S-0h	R-0-0h

Table 15-21. TBSTSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	SYNCI	R-0/W1S	0h	Input Synchronization Latched Status Clear 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TBSTS[SYNCI] bit. Reset type: SYSRSn
0	RESERVED	R-0	0h	Reserved

15.12.2.8 TBCTR Register (Offset = 14h) [Reset = 00000000h]

TBCTR is shown in Figure 15-47 and described in Table 15-22.

Return to the Summary Table.

Time base counter register

Figure 15-47. TBCTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TBCTR															
R-0-0h																R/W-0h															

Table 15-22. TBCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBCTR	R/W	0h	Time Base Counter Register Reset type: SYSRSn

15.12.2.9 CMPCTL Register (Offset = 18h) [Reset = 00000000h]

CMPCTL is shown in Figure 15-48 and described in Table 15-23.

Return to the Summary Table.

Counter compare control register

Figure 15-48. CMPCTL Register

31	30	29	28	27	26	25	24
RESERVED				LOADDMODE		LOADCMODE	
R-0-0h				R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED				PWM3_LOADBMODE		PWM3_LOADAMODE	
R-0-0h				R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED				PWM2_LOADBMODE		PWM2_LOADAMODE	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED				PWM1_LOADBMODE		PWM1_LOADAMODE	
R-0-0h				R/W-0h		R/W-0h	

Table 15-23. CMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	Reserved
27-26	LOADDMODE	R/W	0h	Shadow to Active load of CMPD register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter CCM_CD_PRESENT = 1 Reset type: SYSRSn
25-24	LOADCMODE	R/W	0h	Shadow to Active load of CMPC register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter CCM_CD_PRESENT = 1 Reset type: SYSRSn
23-20	RESERVED	R-0	0h	Reserved
19-18	PWM3_LOADBMODE	R/W	0h	Shadow to Active load of PWM3_CMPB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM3_PRESENT = 1 Reset type: SYSRSn

Table 15-23. CMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17-16	PWM3_LOADAMODE	R/W	0h	Shadow to Active load of PWM3_CMPA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM3_PRESENT = 1 Reset type: SYSRSn
15-12	RESERVED	R-0	0h	Reserved
11-10	PWM2_LOADBMODE	R/W	0h	Shadow to Active load of PWM2_CMPB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM2_PRESENT = 1 Reset type: SYSRSn
9-8	PWM2_LOADAMODE	R/W	0h	Shadow to Active load of PWM2_CMPA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM2_PRESENT = 1 Reset type: SYSRSn
7-4	RESERVED	R-0	0h	Reserved
3-2	PWM1_LOADBMODE	R/W	0h	Shadow to Active load of PWM1_CMPB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
1-0	PWM1_LOADAMODE	R/W	0h	Shadow to Active load of PWM1_CMPA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn

15.12.2.10 CMPC Register (Offset = 20h) [Reset = 00000000h]

CMPC is shown in Figure 15-49 and described in Table 15-24.

Return to the Summary Table.

Counter compare C register

Figure 15-49. CMPC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CMPC															
R-0-0h																R/W-0h															

Table 15-24. CMPC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPC	R/W	0h	Compare C register The value in the CMPC register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare C' event. Shadowing of this register is enabled and disabled by the CMPCTL[LOADCMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.2.11 CMPD Register (Offset = 22h) [Reset = 00000000h]

CMPD is shown in Figure 15-50 and described in Table 15-25.

Return to the Summary Table.

Counter compare D register

Figure 15-50. CMPD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CMPD															
R-0-0h																R/W-0h															

Table 15-25. CMPD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPD	R/W	0h	Compare D register The value in the CMPD register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare D' event. Shadowing of this register is enabled and disabled by the CMPCTL[LOADDMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.2.12 CMPCS Register (Offset = 24h) [Reset = 00000000h]

CMPCS is shown in Figure 15-51 and described in Table 15-26.

Return to the Summary Table.

Counter compare C shadow register

Figure 15-51. CMPCS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CMPCS															
R-0-0h																R/W-0h															

Table 15-26. CMPCS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPCS	R/W	0h	Compare C Shadow Register The value in the CMPCS register is loaded into CMPC register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.13 CMPDS Register (Offset = 26h) [Reset = 00000000h]

CMPDS is shown in Figure 15-52 and described in Table 15-27.

Return to the Summary Table.

Counter compare D shadow register

Figure 15-52. CMPDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CMPDS															
R-0-0h																R/W-0h															

Table 15-27. CMPDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPDS	R/W	0h	Compare D Shadow Register The value in the CMPDS register is loaded into CMPD register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.14 AQCTL Register (Offset = 28h) [Reset = 00000000h]

AQCTL is shown in Figure 15-53 and described in Table 15-28.

Return to the Summary Table.

Action qualifier control register

Figure 15-53. AQCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED				PWM3_LDAQBMODE		PWM3_LDAQAMODE	
R-0-0h				R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED				PWM2_LDAQBMODE		PWM2_LDAQAMODE	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED				PWM1_LDAQBMODE		PWM1_LDAQAMODE	
R-0-0h				R/W-0h		R/W-0h	

Table 15-28. AQCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	Reserved
19-18	PWM3_LDAQBMODE	R/W	0h	Shadow to Active load of PWM3_AQCTLB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM3_PRESENT = 1 Reset type: SYSRSn
17-16	PWM3_LDAQAMODE	R/W	0h	Shadow to Active load of PWM3_AQCTLA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM3_PRESENT = 1 Reset type: SYSRSn
15-12	RESERVED	R-0	0h	Reserved
11-10	PWM2_LDAQBMODE	R/W	0h	Shadow to Active load of PWM2_AQCTLB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM2_PRESENT = 1 Reset type: SYSRSn

Table 15-28. AQCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	PWM2_LDAQAMODE	R/W	0h	Shadow to Active load of PWM2_AQCTLA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM2_PRESENT = 1 Reset type: SYSRSn
7-4	RESERVED	R-0	0h	Reserved
3-2	PWM1_LDAQBMODE	R/W	0h	Shadow to Active load of PWM1_AQCTLB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
1-0	PWM1_LDAQAMODE	R/W	0h	Shadow to Active load of PWM1_AQCTLA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn

15.12.2.15 SOCEN Register (Offset = 30h) [Reset = 00000000h]

SOCEN is shown in Figure 15-54 and described in Table 15-29.

Return to the Summary Table.

Start of conversion enable

Figure 15-54. SOCEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCD_ENABLE	SOCC_ENABLE	SOCB_ENABLE	SOCA_ENABLE
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-29. SOCEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	SOCD_ENABLE	R/W	0h	SOCD Selection Enable 0 - SOCD disable 1- SOCD enable Reset type: SYSRSn
2	SOCC_ENABLE	R/W	0h	SOCC Selection Enable 0 - SOCC disable 1- SOCC enable Reset type: SYSRSn
1	SOCB_ENABLE	R/W	0h	SOCB Selection Enable 0 - SOCB disable 1- SOCB enable Reset type: SYSRSn
0	SOCA_ENABLE	R/W	0h	SOCA Selection Enable 0 - SOCA disable 1- SOCA enable Reset type: SYSRSn

15.12.2.16 SOCSEL Register (Offset = 32h) [Reset = 00000000h]

SOCSEL is shown in Figure 15-55 and described in Table 15-30.

Return to the Summary Table.

Start of conversion selection

Figure 15-55. SOCSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED			SOCD_SEL					RESERVED			SOCC_SEL				
R-0-0h			R/W-0h					R-0-0h			R/W-0h				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			SOCB_SEL					RESERVED			SOCA_SEL				
R-0-0h			R/W-0h					R-0-0h			R/W-0h				

Table 15-30. SOCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R-0	0h	Reserved

Table 15-30. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28-24	SOCD_SEL	R/W	0h	SOCD Selection Options These bits determine when SOCD pulse will be generated. 00000: Reserved (SOC D disabled) 00001: Reserved (SOC D disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC D disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC D disabled) Reset type: SYSRSn
23-21	RESERVED	R-0	0h	Reserved

Table 15-30. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20-16	SOCC_SEL	R/W	0h	SOCC Selection Options These bits determine when SOCC pulse will be generated. 00000: Reserved (SOC C disabled) 00001: Reserved (SOC C disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC C disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC C disabled) Reset type: SYSRSn
15-13	RESERVED	R-0	0h	Reserved

Table 15-30. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-8	SOCB_SEL	R/W	0h	SOCB Selection Options These bits determine when SOCB pulse will be generated. 00000: Reserved (SOC B disabled) 00001: Reserved (SOC B disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC B disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC B disabled) Reset type: SYSRSn
7-5	RESERVED	R-0	0h	Reserved

Table 15-30. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	SOCA_SEL	R/W	0h	SOCA Selection Options These bits determine when SOCA pulse will be generated. 00000: Reserved (SOC A disabled) 00001: Reserved (SOC A disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC A disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC A disabled) Reset type: SYSRSn

15.12.2.17 SOCPERIOD Register (Offset = 34h) [Reset = 00000000h]

SOCPERIOD is shown in Figure 15-56 and described in Table 15-31.

Return to the Summary Table.

Start of conversion period

Figure 15-56. SOCPERIOD Register

31	30	29	28	27	26	25	24
RESERVED					SOCD_PERIOD		
R-0-0h					R/W-0h		
23	22	21	20	19	18	17	16
RESERVED					SOCC_PERIOD		
R-0-0h					R/W-0h		
15	14	13	12	11	10	9	8
RESERVED					SOCB_PERIOD		
R-0-0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					SOCA_PERIOD		
R-0-0h					R/W-0h		

Table 15-31. SOCPERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	Reserved
26-24	SOCD_PERIOD	R/W	0h	SOCD Period These bits determine how many selected SOCSEL[SOCD_SEL] events need to occur before an SOCD pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCD_ENABLE] = 1). The SOCD pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCD] = 1). Once the SOCD pulse is generated, the SOCCNT[SOCD_CNT] bits will automatically be cleared. Reset type: SYSRSn
23-19	RESERVED	R-0	0h	Reserved
18-16	SOCC_PERIOD	R/W	0h	SOCC Period These bits determine how many selected SOCSEL[SOCC_SEL] events need to occur before an SOCC pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCC_ENABLE] = 1). The SOCC pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCC] = 1). Once the SOCC pulse is generated, the SOCCNT[SOCC_CNT] bits will automatically be cleared. Reset type: SYSRSn
15-11	RESERVED	R-0	0h	Reserved
10-8	SOCB_PERIOD	R/W	0h	SOCB Period These bits determine how many selected SOCSEL[SOCB_SEL] events need to occur before an SOCB pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCB_ENABLE] = 1). The SOCB pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCB] = 1). Once the SOCB pulse is generated, the SOCCNT[SOCB_CNT] bits will automatically be cleared. Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved

Table 15-31. SOCPERIOD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	SOCA_PERIOD	R/W	0h	SOCA Period These bits determine how many selected SOCSEL[SOCA_SEL] events need to occur before an SOCA pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCA_ENABLE] = 1). The SOCA pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCA] = 1). Once the SOCA pulse is generated, the SOCCNT[SOCA_CNT] bits will automatically be cleared. Reset type: SYSRSn

15.12.2.18 SOCCNT Register (Offset = 36h) [Reset = 00000000h]

SOCCNT is shown in Figure 15-57 and described in Table 15-32.

Return to the Summary Table.

Start of conversion count

Figure 15-57. SOCCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					SOCD_CNT			RESERVED					SOCC_CNT		
R-0-0h					R-0h			R-0-0h					R-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					SOCB_CNT			RESERVED					SOCA_CNT		
R-0-0h					R-0h			R-0-0h					R-0h		

Table 15-32. SOCCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	Reserved
26-24	SOCD_CNT	R	0h	SOC D Counter Register These bits indicate how many selected SOCSEL[SOCD_SEL] events have occurred. These bits are automatically cleared when a SOCD pulse is generated. Reset type: SYSRSn
23-19	RESERVED	R-0	0h	Reserved
18-16	SOCC_CNT	R	0h	SOC C Counter Register These bits indicate how many selected SOCSEL[SOCC_SEL] events have occurred. These bits are automatically cleared when a SOCC pulse is generated. Reset type: SYSRSn
15-11	RESERVED	R-0	0h	Reserved
10-8	SOCB_CNT	R	0h	SOC B Counter Register These bits indicate how many selected SOCSEL[SOCB_SEL] events have occurred. These bits are automatically cleared when a SOCB pulse is generated. Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	SOCA_CNT	R	0h	SOC A Counter Register These bits indicate how many selected SOCSEL[SOCA_SEL] events have occurred. These bits are automatically cleared when a SOCA pulse is generated. Reset type: SYSRSn

15.12.2.19 SOCFLAG Register (Offset = 38h) [Reset = 00000000h]

SOCFLAG is shown in Figure 15-58 and described in Table 15-33.

Return to the Summary Table.

Start of conversion flag

Figure 15-58. SOCFLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCD	SOCC	SOCB	SOCA
R-0-0h				R-0h	R-0h	R-0h	R-0h

Table 15-33. SOCFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	SOCD	R	0h	Latched SOC D Status Flag SOCD output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCD. The SOCD output will continue to be generated even if the flag bit is set. Reset type: SYSRSn
2	SOCC	R	0h	Latched SOC C Status Flag SOCC output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCC. The SOCC output will continue to be generated even if the flag bit is set. Reset type: SYSRSn
1	SOCB	R	0h	Latched SOC B Status Flag SOCB output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCB. The SOCB output will continue to be generated even if the flag bit is set. Reset type: SYSRSn
0	SOCA	R	0h	Latched SOC A Status Flag SOCA output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCA. The SOCA output will continue to be generated even if the flag bit is set. Reset type: SYSRSn

15.12.2.20 SOCCLR Register (Offset = 3Ah) [Reset = 00000000h]

SOCCLR is shown in Figure 15-59 and described in Table 15-34.

Return to the Summary Table.

Start of conversion clear

Figure 15-59. SOCCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCD	SOCC	SOCB	SOCA
R-0-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 15-34. SOCCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	SOCD	R-0/W1S	0h	Clear SOC D Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCD] bit. Reset type: SYSRSn
2	SOCC	R-0/W1S	0h	Clear SOC C Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCC] bit. Reset type: SYSRSn
1	SOCB	R-0/W1S	0h	Clear SOC B Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCB] bit. Reset type: SYSRSn
0	SOCA	R-0/W1S	0h	Clear SOC A Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCA] bit. Reset type: SYSRSn

15.12.2.21 ETSEL Register (Offset = 40h) [Reset = 00000000h]

ETSEL is shown in Figure 15-60 and described in Table 15-35.

Return to the Summary Table.

Event trigger selection

Figure 15-60. ETSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ET2_SEL					RESERVED			ET1_SEL				
R-0-0h			R/W-0h					R-0-0h			R/W-0h				

Table 15-35. ETSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	Reserved

Table 15-35. ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-8	ET2_SEL	R/W	0h	Event trigger2 Selection Options These bits determine when event trigger pulse will be generated. 00000: Reserved (ET2 Disabled) 00001: Reserved (ET2 Disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (ET2 Disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (ET2 Disabled) Reset type: SYSRSn
7-5	RESERVED	R-0	0h	Reserved

Table 15-35. ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	ET1_SEL	R/W	0h	Event trigger1 Selection Options These bits determine when event trigger pulse will be generated. 00000: Reserved (ET1 Disabled) 00001: Reserved (ET1 Disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (ET1 Disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (ET1 Disabled) Reset type: SYSRSn

15.12.2.22 ETPERIOD Register (Offset = 42h) [Reset = 00000000h]

ETPERIOD is shown in Figure 15-61 and described in Table 15-36.

Return to the Summary Table.

Event trigger period

Figure 15-61. ETPERIOD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED					ET2_PERIOD		
R-0-0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					ET1_PERIOD		
R-0-0h					R/W-0h		

Table 15-36. ETPERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	Reserved
10-8	ET2_PERIOD	R/W	0h	These bits determine how many selected ETSEL[ET2_SEL] events need to occur before an interrupt is generated. If the interrupt status flag is set from a previous interrupt (INTFLG[ET2] = 1) then no interrupt will be generated until the flag is cleared via the INTCLR[ET2] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETCNT[ET2_CNT] bits will automatically be cleared. Writing a PERIOD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a PERIOD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero PERIOD value is written, the counter is incremented. Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	ET1_PERIOD	R/W	0h	These bits determine how many selected ETSEL[ET1_SEL] events need to occur before an interrupt is generated. If the interrupt status flag is set from a previous interrupt (INTFLG[ET1] = 1) then no interrupt will be generated until the flag is cleared via the INTCLR[ET1] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETCNT[ET1_CNT] bits will automatically be cleared. Writing a PERIOD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a PERIOD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero PERIOD value is written, the counter is incremented. Reset type: SYSRSn

15.12.2.23 ETCNT Register (Offset = 44h) [Reset = 00000000h]

ETCNT is shown in Figure 15-62 and described in Table 15-37.

Return to the Summary Table.

Event trigger count

Figure 15-62. ETCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					ET2_CNT			RESERVED					ET1_CNT		
R-0-0h					R-0h			R-0-0h					R-0h		

Table 15-37. ETCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	Reserved
10-8	ET2_CNT	R	0h	Event trigger2 Counter Register These bits indicate how many selected ET_SEL[ET2_SEL] events have occurred. These bits are automatically cleared once INTFLAG.ET2 is generated (irrespective of INTEN.ET2 configuration). Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	ET1_CNT	R	0h	Event trigger1 Counter Register These bits indicate how many selected ET_SEL[ET1_SEL] events have occurred. These bits are automatically cleared once INTFLAG.ET1 is generated (irrespective of INTEN.ET1 configuration). Reset type: SYSRSn

15.12.2.24 INTEN Register (Offset = 48h) [Reset = 00000000h]

INTEN is shown in Figure 15-63 and described in Table 15-38.

Return to the Summary Table.

Interrupt Enable Register

Figure 15-63. INTEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	RESERVED
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 15-38. INTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R/W	0h	Counter Overflow Interrupt Enable 1 : Enables CNT_OVF interrupt 0 : Disable CNT_OVF interrupt Reset type: SYSRSn
4	ET2	R/W	0h	Event Trigger 2 Interrupt Enable 1 : Enables ET2 interrupt 0 : Disable ET2 interrupt Reset type: SYSRSn
3	ET1	R/W	0h	Event Trigger 1 Interrupt Enable 1 : Enables ET1 interrupt 0 : Disable ET1 interrupt Reset type: SYSRSn
2	OST	R/W	0h	Trip-zone One-Shot Interrupt Enable 1 : Enables OST interrupt 0 : Disable OST interrupt Reset type: SYSRSn
1	CBC	R/W	0h	Trip-zone Cycle-by-Cycle Interrupt Enable 1 : Enables CBC interrupt 0 : Disable CBC interrupt Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

15.12.2.25 INTFLAG Register (Offset = 4Ah) [Reset = 00000000h]

INTFLAG is shown in Figure 15-64 and described in Table 15-39.

Return to the Summary Table.

Interrupt Flag Register

Figure 15-64. INTFLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	INT
R-0-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-39. INTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R	0h	Latched Status Flag for A Counter Overflow Event 1 : CNT_OVF flag is set 0 : CNT_OVF flag is not set Reset type: SYSRSn
4	ET2	R	0h	Latched Status Flag for A Event Trigger 2 Event 1: ET2 flag is set 0: ET2 flag is not set Reset type: SYSRSn
3	ET1	R	0h	Latched Status Flag for A Event Trigger 1 Event 1: ET1 flag is set 0: ET1 flag is not set Reset type: SYSRSn
2	OST	R	0h	Latched Status Flag for A One-Shot Trip Event 1: OST flag is set 0: OST flag is not set Reset type: SYSRSn
1	CBC	R	0h	Latched Status Flag for Cycle-By-Cycle Trip Event 1: CBC flag is set 0: CBC flag is not set Reset type: SYSRSn
0	INT	R	0h	Global Interrupt Status Flag 1: Global flag is set 0: Global flag is not set Reset type: SYSRSn

15.12.2.26 INTCLR Register (Offset = 4Ch) [Reset = 00000000h]

INTCLR is shown in Figure 15-65 and described in Table 15-40.

Return to the Summary Table.

Interrupt Clear Register

Figure 15-65. INTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	INT
R-0-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 15-40. INTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R-0/W1S	0h	Clear Counter Overflow Flag Writing '1' will clear INTFLAG[CNT_OVF] register. Reset type: SYSRSn
4	ET2	R-0/W1S	0h	Clear Event Trigger 2 Flag Writing '1' will clear INTFLAG[ET2] register. Reset type: SYSRSn
3	ET1	R-0/W1S	0h	Clear Event Trigger 1 Flag Writing '1' will clear INTFLAG[ET1] register. Reset type: SYSRSn
2	OST	R-0/W1S	0h	Clear One-Shot Trip Latch Writing '1' will clear INTFLAG[OST] register. Reset type: SYSRSn
1	CBC	R-0/W1S	0h	Clear Cycle-by-Cycle Trip Latch Writing '1' will clear INTFLAG[CBC] register. Reset type: SYSRSn
0	INT	R-0/W1S	0h	Clear Global Interrupt Flag Writing '1' will clear INTFLAG[INT] register. Reset type: SYSRSn

15.12.2.27 INTFRC Register (Offset = 4Eh) [Reset = 00000000h]

INTFRC is shown in Figure 15-66 and described in Table 15-41.

Return to the Summary Table.

Interrupt Force Register

Figure 15-66. INTFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	RESERVED
R-0-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 15-41. INTFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R-0/W1S	0h	Force Counter Overflow Interrupt Writing '1' will set INTFLAG[CNT_OVF] register. Reset type: SYSRSn
4	ET2	R-0/W1S	0h	Force Event Trigger 2 Interrupt Writing '1' will set INTFLAG[ET2] register. Reset type: SYSRSn
3	ET1	R-0/W1S	0h	Force Event Trigger 1 Interrupt Writing '1' will set INTFLAG[ET1] register. Reset type: SYSRSn
2	OST	R-0/W1S	0h	Force One-Shot Trip Interrupt Writing '1' will set INTFLAG[OST] register. Reset type: SYSRSn
1	CBC	R-0/W1S	0h	Force Cycle-by-Cycle Trip Interrupt Writing '1' will set INTFLAG[CBC] register. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

15.12.2.28 TZSEL Register (Offset = 50h) [Reset = 00000000h]

TZSEL is shown in Figure 15-67 and described in Table 15-42.

Return to the Summary Table.

Trip Zone Selection

Figure 15-67. TZSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
OST8	OST7	OST6	OST5	OST4	OST3	OST2	OST1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CBC8	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-42. TZSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	OST8	R/W	0h	Select Trip-zone 8 (TZ8) for OST generation 0: Disable TZ8 as a OST trip source for this MCPWM module 1: Enable TZ8 as a OST trip source for this MCPWM module Reset type: SYSRSn
22	OST7	R/W	0h	Select Trip-zone 7 (TZ7) for OST generation 0: Disable TZ7 as a OST trip source for this MCPWM module 1: Enable TZ7 as a OST trip source for this MCPWM module Reset type: SYSRSn
21	OST6	R/W	0h	Select Trip-zone 6 (TZ6) for OST generation 0: Disable TZ6 as a OST trip source for this MCPWM module 1: Enable TZ6 as a OST trip source for this MCPWM module Reset type: SYSRSn
20	OST5	R/W	0h	Select Trip-zone 5 (TZ5) for OST generation 0: Disable TZ5 as a OST trip source for this MCPWM module 1: Enable TZ5 as a OST trip source for this MCPWM module Reset type: SYSRSn
19	OST4	R/W	0h	Select Trip-zone 4 (TZ4) for OST generation 0: Disable TZ4 as a OST trip source for this MCPWM module 1: Enable TZ4 as a OST trip source for this MCPWM module Reset type: SYSRSn
18	OST3	R/W	0h	Select Trip-zone 3 (TZ3) for OST generation 0: Disable TZ3 as a OST trip source for this MCPWM module 1: Enable TZ3 as a OST trip source for this MCPWM module Reset type: SYSRSn
17	OST2	R/W	0h	Select Trip-zone 2 (TZ2) for OST generation 0: Disable TZ2 as a OST trip source for this MCPWM module 1: Enable TZ2 as a OST trip source for this MCPWM module Reset type: SYSRSn

Table 15-42. TZSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	OST1	R/W	0h	Select Trip-zone 1 (TZ1) for OST generation 0: Disable TZ1 as a OST trip source for this MCPWM module 1: Enable TZ1 as a OST trip source for this MCPWM module Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7	CBC8	R/W	0h	Select Trip-zone 8 (TZ8) for CBC generation 0: Disable TZ8 as a CBC trip source for this MCPWM module 1: Enable TZ8 as a CBC trip source for this MCPWM module Reset type: SYSRSn
6	CBC7	R/W	0h	Select Trip-zone 7 (TZ7) for CBC generation 0: Disable TZ7 as a CBC trip source for this MCPWM module 1: Enable TZ7 as a CBC trip source for this MCPWM module Reset type: SYSRSn
5	CBC6	R/W	0h	Select Trip-zone 6 (TZ6) for CBC generation 0: Disable TZ6 as a CBC trip source for this MCPWM module 1: Enable TZ6 as a CBC trip source for this MCPWM module Reset type: SYSRSn
4	CBC5	R/W	0h	Select Trip-zone 5 (TZ5) for CBC generation 0: Disable TZ5 as a CBC trip source for this MCPWM module 1: Enable TZ5 as a CBC trip source for this MCPWM module Reset type: SYSRSn
3	CBC4	R/W	0h	Select Trip-zone 4 (TZ4) for CBC generation 0: Disable TZ4 as a CBC trip source for this MCPWM module 1: Enable TZ4 as a CBC trip source for this MCPWM module Reset type: SYSRSn
2	CBC3	R/W	0h	Select Trip-zone 3 (TZ3) for CBC generation 0: Disable TZ3 as a CBC trip source for this MCPWM module 1: Enable TZ3 as a CBC trip source for this MCPWM module Reset type: SYSRSn
1	CBC2	R/W	0h	Select Trip-zone 2 (TZ2) for CBC generation 0: Disable TZ2 as a CBC trip source for this MCPWM module 1: Enable TZ2 as a CBC trip source for this MCPWM module Reset type: SYSRSn
0	CBC1	R/W	0h	Select Trip-zone 1 (TZ1) for CBC generation 0: Disable TZ1 as a CBC trip source for this MCPWM module 1: Enable TZ1 as a CBC trip source for this MCPWM module Reset type: SYSRSn

15.12.2.29 TZCTL Register (Offset = 56h) [Reset = 00000010h]

TZCTL is shown in Figure 15-68 and described in Table 15-43.

Return to the Summary Table.

Trip Zone control

Figure 15-68. TZCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CBCPULSE		TZB		TZA	
R-0-0h		R/W-1h		R/W-0h		R/W-0h	

Table 15-43. TZCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-4	CBCPULSE	R/W	1h	Clear Pulse for Cycle-By-Cycle (CBC) Trip Latch This bit field determines which pulse clears the CBC trip latch. 00: CBC trip latch is not cleared 01: CTR = ZERO pulse clears CBC trip latch 10: CTR = PRD pulse clears CBC trip latch 11: CTR = ZERO or CTR = PRD pulse clears CBC trip latch Reset type: SYSRSn
3-2	TZB	R/W	0h	Trip action on PWMxB 00: High-impedance (PWMxB = High-impedance state) 01: Force PWMxB to a high state 10: Force PWMxB to a low state 11: Do nothing, no action is taken on PWMxB. Reset type: SYSRSn
1-0	TZA	R/W	0h	Trip action on PWMxA 00: High-impedance (PWMxA = High-impedance state) 01: Force PWMxA to a high state 10: Force PWMxA to a low state 11: Do nothing, no action is taken on PWMxA. Reset type: SYSRSn

15.12.2.30 TZCBCOSTFLAG Register (Offset = 58h) [Reset = 00000000h]

TZCBCOSTFLAG is shown in Figure 15-69 and described in Table 15-44.

Return to the Summary Table.

Trip Zone CBC and OST Flag Register

Figure 15-69. TZCBCOSTFLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
OST8	OST7	OST6	OST5	OST4	OST3	OST2	OST1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CBC8	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-44. TZCBCOSTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	OST8	R	0h	Latched Status Flag for OST TZ8 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ8. 1: Reading a 1 indicates a OST trip has occured by TZ8. Reset type: SYSRSn
22	OST7	R	0h	Latched Status Flag for OST TZ7 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ7. 1: Reading a 1 indicates a OST trip has occured by TZ7. Reset type: SYSRSn
21	OST6	R	0h	Latched Status Flag for OST TZ6 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ6. 1: Reading a 1 indicates a OST trip has occured by TZ6. Reset type: SYSRSn
20	OST5	R	0h	Latched Status Flag for OST TZ5 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ5. 1: Reading a 1 indicates a OST trip has occured by TZ5. Reset type: SYSRSn
19	OST4	R	0h	Latched Status Flag for OST TZ4 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ4. 1: Reading a 1 indicates a OST trip has occured by TZ4. Reset type: SYSRSn
18	OST3	R	0h	Latched Status Flag for OST TZ3 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ3. 1: Reading a 1 indicates a OST trip has occured by TZ3. Reset type: SYSRSn
17	OST2	R	0h	Latched Status Flag for OST TZ2 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ2. 1: Reading a 1 indicates a OST trip has occured by TZ2. Reset type: SYSRSn

Table 15-44. TZCBCOSTFLAG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	OST1	R	0h	Latched Status Flag for OST TZ1 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ1. 1: Reading a 1 indicates a OST trip has occured by TZ1. Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7	CBC8	R	0h	Latched Status Flag for CBC TZ8 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ8. 1: Reading a 1 indicates a CBC trip has occured by TZ8. Reset type: SYSRSn
6	CBC7	R	0h	Latched Status Flag for CBC TZ7 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ7. 1: Reading a 1 indicates a CBC trip has occured by TZ7. Reset type: SYSRSn
5	CBC6	R	0h	Latched Status Flag for CBC TZ6 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ6. 1: Reading a 1 indicates a CBC trip has occured by TZ6. Reset type: SYSRSn
4	CBC5	R	0h	Latched Status Flag for CBC TZ5 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ5. 1: Reading a 1 indicates a CBC trip has occured by TZ5. Reset type: SYSRSn
3	CBC4	R	0h	Latched Status Flag for CBC TZ4 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ4. 1: Reading a 1 indicates a CBC trip has occured by TZ4. Reset type: SYSRSn
2	CBC3	R	0h	Latched Status Flag for CBC TZ3 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ3. 1: Reading a 1 indicates a CBC trip has occured by TZ3. Reset type: SYSRSn
1	CBC2	R	0h	Latched Status Flag for CBC TZ2 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ2. 1: Reading a 1 indicates a CBC trip has occured by TZ2. Reset type: SYSRSn
0	CBC1	R	0h	Latched Status Flag for CBC TZ1 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ1. 1: Reading a 1 indicates a CBC trip has occured by TZ1. Reset type: SYSRSn

15.12.2.31 TZCBCOSTCLR Register (Offset = 5Ah) [Reset = 00000000h]

TZCBCOSTCLR is shown in Figure 15-70 and described in Table 15-45.

Return to the Summary Table.

Trip Zone CBC and OST Clear Register

Figure 15-70. TZCBCOSTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
OST8	OST7	OST6	OST5	OST4	OST3	OST2	OST1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CBC8	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 15-45. TZCBCOSTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	OST8	R-0/W1S	0h	Clear OST TZ8 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST8] bit. Reset type: SYSRSn
22	OST7	R-0/W1S	0h	Clear OST TZ7 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST7] bit. Reset type: SYSRSn
21	OST6	R-0/W1S	0h	Clear OST TZ6 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST6] bit. Reset type: SYSRSn
20	OST5	R-0/W1S	0h	Clear OST TZ5 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST5] bit. Reset type: SYSRSn
19	OST4	R-0/W1S	0h	Clear OST TZ4 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST4] bit. Reset type: SYSRSn
18	OST3	R-0/W1S	0h	Clear OST TZ3 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST3] bit. Reset type: SYSRSn
17	OST2	R-0/W1S	0h	Clear OST TZ2 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST2] bit. Reset type: SYSRSn

Table 15-45. TZCBCOSTCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	OST1	R-0/W1S	0h	Clear OST TZ1 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST1] bit. Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7	CBC8	R-0/W1S	0h	Clear CBC TZ8 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC8] bit. Reset type: SYSRSn
6	CBC7	R-0/W1S	0h	Clear CBC TZ7 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC7] bit. Reset type: SYSRSn
5	CBC6	R-0/W1S	0h	Clear CBC TZ6 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC6] bit. Reset type: SYSRSn
4	CBC5	R-0/W1S	0h	Clear CBC TZ5 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC5] bit. Reset type: SYSRSn
3	CBC4	R-0/W1S	0h	Clear CBC TZ4 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC4] bit. Reset type: SYSRSn
2	CBC3	R-0/W1S	0h	Clear CBC TZ3 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC3] bit. Reset type: SYSRSn
1	CBC2	R-0/W1S	0h	Clear CBC TZ2 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC2] bit. Reset type: SYSRSn
0	CBC1	R-0/W1S	0h	Clear CBC TZ1 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC1] bit. Reset type: SYSRSn

15.12.2.32 DBCTL Register (Offset = 60h) [Reset = 00000000h]

DBCTL is shown in Figure 15-71 and described in Table 15-46.

Return to the Summary Table.

Dead band control register

Figure 15-71. DBCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED				LOADREDMODE		LOADFEDMODE	
R-0-0h				R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED							DEDB_MODE
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
OUTSWAP		IN_MODE		POLSEL		OUT_MODE	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-46. DBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	Reserved
19-18	LOADREDMODE	R/W	0h	Shadow to Active load of DBRED register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
17-16	LOADFEDMODE	R/W	0h	Shadow to Active load of DBFED register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
15-9	RESERVED	R-0	0h	Reserved
8	DEDB_MODE	R/W	0h	Dead Band Dual-Edge B Mode Control (S8 switch) 0: Rising edge delay applied to InA/InB as selected by S4 switch (IN-MODE bits) on A signal path only. Falling edge delay applied to InA/InB as selected by S5 switch (INMODE bits) on B signal path only. 1: Rising edge delay and falling edge delay applied to source selected by S4 switch (INMODE bits) and output to B signal path only. Note: When this bit is set to 1, user should always either set OUT_MODE bits such that Apath = InA OR OUTSWAP bits such that OutA=Bpath otherwise, OutA will be invalid. Reset type: SYSRSn

Table 15-46. DBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	OUTSWAP	R/W	0h	Dead Band Output Swap Control Bit 7 controls the S6 switch and bit 6 controls the S7 switch. 00: OutA and OutB signals are as defined by OUT-MODE bits. 01: OutA = A-path as defined by OUT-MODE bits. OutB = A-path as defined by OUT-MODE bits (rising edge delay or delay-bypassed A signal path). 10: OutA = B-path as defined by OUT-MODE bits (falling edge delay or delay-bypassed B signal path). OutB = B-path as defined by OUT-MODE bits. 11: OutA = B-path as defined by OUT-MODE bits (falling edge delay or delay-bypassed B signal path). OutB = A-path as defined by OUT-MODE bits (rising edge delay or delay-bypassed A signal path). Reset type: SYSRSn
5-4	IN_MODE	R/W	0h	Dead-Band Input Mode Control Bit 5 controls the S5 switch and bit 4 controls the S4 switch shown. This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms the default is PWMxA In is the source for both falling and rising-edge delays. 00: PWMxA In (from the action-qualifier) is the source for both falling-edge and rising-edge delay. 01: PWMxB In (from the action-qualifier) is the source for rising-edge delayed signal. PWMxA In (from the action-qualifier) is the source for falling-edge delayed signal. 10: PWMxA In (from the action-qualifier) is the source for rising-edge delayed signal. PWMxB In (from the action-qualifier) is the source for falling-edge delayed signal. 11: PWMxB In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal. Reset type: SYSRSn
3-2	POLSEL	R/W	0h	Polarity Select Control Bit 3 controls the S3 switch and bit 2 controls the S2 switch. This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule. The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that DBCTL[OUT_MODE] = 1,1 and DBCTL[IN_MODE] = 0x0. Other enhanced modes are also possible, but not regarded as typical usage modes. 00: Active high (AH) mode. Neither PWMxA nor PWMxB is inverted (default). 01: Active low complementary (ALC) mode. PWMxA is inverted. 10: Active high complementary (AHC). PWMxB is inverted. 11: Active low (AL) mode. Both PWMxA and PWMxB are inverted. Reset type: SYSRSn
1-0	OUT_MODE	R/W	0h	Dead-Band Output Mode Control Bit 1 controls the S1 switch and bit 0 controls the S0 switch. 00: DBM is fully disabled or by-passed. In this mode the POLSEL and IN-MODE bits have no effect. 01: Apath = InA (delay is by-passed for A signal path) Bpath = FED (Falling Edge Delay in B signal path) 10: Apath = RED (Rising Edge Delay in A signal path) Bpath = InB (delay is by-passed for B signal path) 11: DBM is fully enabled (i.e. both RED and FED active) Reset type: SYSRSn

15.12.2.33 DBFED Register (Offset = 68h) [Reset = 00000000h]

DBFED is shown in Figure 15-72 and described in Table 15-47.

Return to the Summary Table.

Dead band fall edge delay

Figure 15-72. DBFED Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		DBFED													
R-0-0h																		R/W-0h													

Table 15-47. DBFED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBFED	R/W	0h	Falling Edge Delay Count 14-bit counter Reset type: SYSRSn

15.12.2.34 DBRED Register (Offset = 6Ah) [Reset = 00000000h]

DBRED is shown in Figure 15-73 and described in Table 15-48.

Return to the Summary Table.

Dead band rise edge delay

Figure 15-73. DBRED Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		DBRED													
R-0-0h																		R/W-0h													

Table 15-48. DBRED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBRED	R/W	0h	Rising Edge Delay Count 14-bit counter Reset type: SYSRSn

15.12.2.35 DBFEDS Register (Offset = 6Ch) [Reset = 00000000h]

DBFEDS is shown in Figure 15-74 and described in Table 15-49.

Return to the Summary Table.

Dead band fall edge delay shadow register

Figure 15-74. DBFEDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		DBFEDS													
R-0-0h																		R/W-0h													

Table 15-49. DBFEDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBFEDS	R/W	0h	DBFED Shadow Register The value in the DBFEDS register is loaded into DBFED register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.36 DBREDS Register (Offset = 6Eh) [Reset = 00000000h]

DBREDS is shown in Figure 15-75 and described in Table 15-50.

Return to the Summary Table.

Dead band rise edge delay shadow register

Figure 15-75. DBREDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		DBREDS													
R-0-0h																		R/W-0h													

Table 15-50. DBREDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBREDS	R/W	0h	DBRED Shadow Register The value in the DBREDS register is loaded into DBRED register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.37 GLDCTL Register (Offset = 78h) [Reset = 00000000h]

GLDCTL is shown in Figure 15-76 and described in Table 15-51.

Return to the Summary Table.

Global load control register

Figure 15-76. GLDCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		GLDMODE		RESERVED		OSHTMODE	GLD
R-0-0h		R/W-0h		R-0-0h		R/W-0h	R/W-0h

Table 15-51. GLDCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-4	GLDMODE	R/W	0h	Select global load event 00: CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: CTR = Zero or CTR = PRD 11: GLDOSHTCTL[GFRCLD] - Softwrae load Reset type: SYSRSn
3-2	RESERVED	R-0	0h	Reserved
1	OSHTMODE	R/W	0h	Global load one-shot enable 0: Disable global load one-shot 1: Enable global load one-shot Reset type: SYSRSn
0	GLD	R/W	0h	Global load enable 0: Disable global load 1: Enable global load Reset type: SYSRSn

15.12.2.38 GLDOSHTCTL Register (Offset = 7Ah) [Reset = 00000000h]

GLDOSHTCTL is shown in Figure 15-77 and described in Table 15-52.

Return to the Summary Table.

Global load one shot control register

Figure 15-77. GLDOSHTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					GFRCLD	OSHTCLR	OSHTLD
R-0-0h					R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 15-52. GLDOSHTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	GFRCLD	R-0/W1S	0h	Force Load Event in One Shot Mode 0: Writing of 0 will be ignored. Always reads back a 0. 1: Force load event. This bit is intended to be used for testing and/or software force loading of the events in global load mode. Reset type: SYSRSn
1	OSHTCLR	R-0/W1S	0h	Clear One Shot latch 0: Writing of 0 will be ignored. Always reads back a 0. 1: Turns the one shot latch condition OFF. Reset type: SYSRSn
0	OSHTLD	R-0/W1S	0h	Enable Reload Event in One Shot Mode 0: Writing of 0 will be ignored. Always reads back a 0. 1: Turns the one shot latch condition ON. Upon occurrence of a chosen load strobe, one shadow to active reload occurs and the latch will be cleared. Hence writing 1 to this bit would allow one load strobe event to pass through and block further strobe events. Reset type: SYSRSn

15.12.2.39 GLDOSHTSTS Register (Offset = 7Ch) [Reset = 00000000h]

GLDOSHTSTS is shown in Figure 15-78 and described in Table 15-53.

Return to the Summary Table.

Global load one shot status register

Figure 15-78. GLDOSHTSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							OSHTLATCH
R-0-0h							R-0h

Table 15-53. GLDOSHTSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	OSHTLATCH	R	0h	One shot latch output 0: one shot latch condition is OFF. 1: one shot latch condition is ON. Reset type: SYSRSn

15.12.2.40 PWM1_CMPA Register (Offset = 80h) [Reset = 00000000h]

PWM1_CMPA is shown in Figure 15-79 and described in Table 15-54.

Return to the Summary Table.

PWM1 counter compare A register

Figure 15-79. PWM1_CMPA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPA															
R-0-0h																R/W-0h															

Table 15-54. PWM1_CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPA	R/W	0h	PWM1 Compare A register The value in the PWM1_CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM1 counter-compare module generates a 'time-base counter equal to CMP A' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM1A or the PWM1B output depending on the configuration of the PWM1_AQCTLA and PWM1_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM1_LOADAMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.2.41 PWM1_CMPAS Register (Offset = 82h) [Reset = 00000000h]

PWM1_CMPAS is shown in Figure 15-80 and described in Table 15-55.

Return to the Summary Table.

PWM1 counter compare A shadow register

Figure 15-80. PWM1_CMPAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPAS															
R-0-0h																R/W-0h															

Table 15-55. PWM1_CMPAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPAS	R/W	0h	PWM1 Compare A Shadow Register The value in the PWM1_CMPAS register is loaded into PWM!_CMPA register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.42 PWM1_CMPB Register (Offset = 84h) [Reset = 00000000h]

PWM1_CMPB is shown in Figure 15-81 and described in Table 15-56.

Return to the Summary Table.

PWM1 counter compare B register

Figure 15-81. PWM1_CMPB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPB															
R-0-0h																R/W-0h															

Table 15-56. PWM1_CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPB	R/W	0h	PWM1 Compare B register The value in the PWM1_CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM1 counter-compare module generates a 'time-base counter equal to CMP B' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM1A or the PWM1B output depending on the configuration of the PWM1_AQCTLA and PWM1_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM1_LOADBMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.2.43 PWM1_CMPBS Register (Offset = 86h) [Reset = 00000000h]

PWM1_CMPBS is shown in Figure 15-82 and described in Table 15-57.

Return to the Summary Table.

PWM1 counter compare B shadow register

Figure 15-82. PWM1_CMPBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPBS															
R-0-0h																R/W-0h															

Table 15-57. PWM1_CMPBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPBS	R/W	0h	PWM1 Compare B Shadow Register The value in the PWM1 CMPBS register is loaded into PWM1 CMPB register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.44 PWM1_AQCTLA Register (Offset = 90h) [Reset = 00000000h]

PWM1_AQCTLA is shown in Figure 15-83 and described in Table 15-58.

Return to the Summary Table.

PWM1 action qualifier A register

Figure 15-83. PWM1_AQCTLA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-58. PWM1_AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-58. PWM1_AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.45 PWM1_AQCTLAS Register (Offset = 92h) [Reset = 00000000h]

PWM1_AQCTLAS is shown in Figure 15-84 and described in Table 15-59.

Return to the Summary Table.

PWM1 action qualifier A shadow register

Figure 15-84. PWM1_AQCTLAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-59. PWM1_AQCTLAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-59. PWM1_AQCTLAS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.46 PWM1_AQCTLB Register (Offset = 94h) [Reset = 00000000h]

PWM1_AQCTLB is shown in Figure 15-85 and described in Table 15-60.

Return to the Summary Table.

PWM1 action qualifier B register

Figure 15-85. PWM1_AQCTLB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-60. PWM1_AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-60. PWM1_AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.47 PWM1_AQCTLBS Register (Offset = 96h) [Reset = 00000000h]

PWM1_AQCTLBS is shown in Figure 15-86 and described in Table 15-61.

Return to the Summary Table.

PWM1 action qualifier B shadow register

Figure 15-86. PWM1_AQCTLBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-61. PWM1_AQCTLBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-61. PWM1_AQCTLBS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.48 PWM1_AQSFRC Register (Offset = 98h) [Reset = 00000000h]

PWM1_AQSFRC is shown in Figure 15-87 and described in Table 15-62.

Return to the Summary Table.

PWM1 action qualifier software force

Figure 15-87. PWM1_AQSFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	PWMB			RESERVED	PWMA		
R-0-0h	R/W-0h			R-0-0h	R/W-0h		

Table 15-62. PWM1_AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6-4	PWMB	R/W	0h	Action qualifier software force on PWMB 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output B 010: Forces a continuous high on output B 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM1_AQOTSFRC[PWMB] = '1'. 110: Set (high) when PWM1_AQOTSFRC[PWMB] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM1_AQOTSFRC[PWMB] = '1'. Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved
2-0	PWMA	R/W	0h	Action qualifier software force on PWMA 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output A 010: Forces a continuous high on output A 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM1_AQOTSFRC[PWMA] = '1'. 110: Set (high) when PWM1_AQOTSFRC[PWMA] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM1_AQOTSFRC[PWMA] = '1'. Reset type: SYSRSn

15.12.2.49 PWM1_AQOTSFRC Register (Offset = 9Ah) [Reset = 00000000h]

PWM1_AQOTSFRC is shown in Figure 15-88 and described in Table 15-63.

Return to the Summary Table.

PWM1 action qualifier one time software force

Figure 15-88. PWM1_AQOTSFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			PWMB	RESERVED			PWMA
R-0-0h			R-0/W1S-0h	R-0-0h			R-0/W1S-0h

Table 15-63. PWM1_AQOTSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4	PWMB	R-0/W1S	0h	Action qualifier one time software force on PWMB 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMB. 1: Initiates a single software forced event Reset type: SYSRSn
3-1	RESERVED	R-0	0h	Reserved
0	PWMA	R-0/W1S	0h	Action qualifier one time software force on PWMA 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMA. 1: Initiates a single software forced event Reset type: SYSRSn

15.12.2.50 PWM2_CMPA Register (Offset = 180h) [Reset = 00000000h]

PWM2_CMPA is shown in Figure 15-89 and described in Table 15-64.

Return to the Summary Table.

PWM2 counter compare A register

Figure 15-89. PWM2_CMPA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM2_CMPA															
R-0-0h																R/W-0h															

Table 15-64. PWM2_CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM2_CMPA	R/W	0h	PWM2 Compare A register The value in the PWM2_CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM2 counter-compare module generates a 'time-base counter equal to CMP A' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM2A or the PWM2B output depending on the configuration of the PWM2_AQCTLA and PWM2_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM2_LOADAMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.2.51 PWM2_CMPAS Register (Offset = 182h) [Reset = 00000000h]

PWM2_CMPAS is shown in Figure 15-90 and described in Table 15-65.

Return to the Summary Table.

PWM2 counter compare A shadow register

Figure 15-90. PWM2_CMPAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM2_CMPAS															
R-0-0h																R/W-0h															

Table 15-65. PWM2_CMPAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM2_CMPAS	R/W	0h	PWM2 Compare A Shadow Register The value in the PWM2_CMPAS register is loaded into PWM2_CMPA register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.52 PWM2_CMPB Register (Offset = 184h) [Reset = 00000000h]

PWM2_CMPB is shown in Figure 15-91 and described in Table 15-66.

Return to the Summary Table.

PWM2 counter compare B register

Figure 15-91. PWM2_CMPB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM2_CMPB															
R-0-0h																R/W-0h															

Table 15-66. PWM2_CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM2_CMPB	R/W	0h	PWM2 Compare B register The value in the PWM2_CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM2 counter-compare module generates a 'time-base counter equal to CMP B' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM2A or the PWM2B output depending on the configuration of the PWM2_AQCTLA and PWM2_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM2_LOADBMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.2.53 PWM2_CMPBS Register (Offset = 186h) [Reset = 00000000h]

PWM2_CMPBS is shown in Figure 15-92 and described in Table 15-67.

Return to the Summary Table.

PWM2 counter compare B shadow register

Figure 15-92. PWM2_CMPBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM2_CMPBS															
R-0-0h																R/W-0h															

Table 15-67. PWM2_CMPBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM2_CMPBS	R/W	0h	PWM2 Compare B Shadow Register The value in the PWM2_CMPBS register is loaded into PWM2_CMPB register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.54 PWM2_AQCTLA Register (Offset = 190h) [Reset = 00000000h]

PWM2_AQCTLA is shown in Figure 15-93 and described in Table 15-68.

Return to the Summary Table.

PWM2 action qualifier A register

Figure 15-93. PWM2_AQCTLA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-68. PWM2_AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM2_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM2_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM2_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM2_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-68. PWM2_AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.55 PWM2_AQCTLAS Register (Offset = 192h) [Reset = 00000000h]

PWM2_AQCTLAS is shown in Figure 15-94 and described in Table 15-69.

Return to the Summary Table.

PWM2 action qualifier A shadow register

Figure 15-94. PWM2_AQCTLAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-69. PWM2_AQCTLAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM2_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM2_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM2_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM2_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-69. PWM2_AQCTLAS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.56 PWM2_AQCTLB Register (Offset = 194h) [Reset = 00000000h]

PWM2_AQCTLB is shown in Figure 15-95 and described in Table 15-70.

Return to the Summary Table.

PWM2 action qualifier B register

Figure 15-95. PWM2_AQCTLB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-70. PWM2_AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM2_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM2_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM2_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM2_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-70. PWM2_AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.57 PWM2_AQCTLBS Register (Offset = 196h) [Reset = 00000000h]

PWM2_AQCTLBS is shown in Figure 15-96 and described in Table 15-71.

Return to the Summary Table.

PWM2 action qualifier B shadow register

Figure 15-96. PWM2_AQCTLBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-71. PWM2_AQCTLBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM2_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM2_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM2_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM2_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-71. PWM2_AQCTLBS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.58 PWM2_AQSFRC Register (Offset = 198h) [Reset = 00000000h]

PWM2_AQSFRC is shown in Figure 15-97 and described in Table 15-72.

Return to the Summary Table.

PWM2 action qualifier software force

Figure 15-97. PWM2_AQSFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	PWMB			RESERVED	PWMA		
R-0-0h	R/W-0h			R-0-0h	R/W-0h		

Table 15-72. PWM2_AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6-4	PWMB	R/W	0h	Action qualifier software force on PWMB 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output B 010: Forces a continuous high on output B 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM2_AQOTSFRC[PWMB] = '1'. 110: Set (high) when PWM2_AQOTSFRC[PWMB] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM2_AQOTSFRC[PWMB] = '1'. Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved
2-0	PWMA	R/W	0h	Action qualifier software force on PWMA 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output A 010: Forces a continuous high on output A 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM2_AQOTSFRC[PWMA] = '1'. 110: Set (high) when PWM2_AQOTSFRC[PWMA] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM2_AQOTSFRC[PWMA] = '1'. Reset type: SYSRSn

15.12.2.59 PWM2_AQOTSFRC Register (Offset = 19Ah) [Reset = 00000000h]

PWM2_AQOTSFRC is shown in Figure 15-98 and described in Table 15-73.

Return to the Summary Table.

PWM2 action qualifier one time software force

Figure 15-98. PWM2_AQOTSFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			PWMB	RESERVED			PWMA
R-0-0h			R-0/W1S-0h	R-0-0h			R-0/W1S-0h

Table 15-73. PWM2_AQOTSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4	PWMB	R-0/W1S	0h	Action qualifier one time software force on PWMB 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMB. 1: Initiates a single software forced event Reset type: SYSRSn
3-1	RESERVED	R-0	0h	Reserved
0	PWMA	R-0/W1S	0h	Action qualifier one time software force on PWMA 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMA. 1: Initiates a single software forced event Reset type: SYSRSn

15.12.2.60 PWM3_CMPA Register (Offset = 280h) [Reset = 00000000h]

PWM3_CMPA is shown in Figure 15-99 and described in Table 15-74.

Return to the Summary Table.

PWM3 counter compare A register

Figure 15-99. PWM3_CMPA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM3_CMPA															
R-0-0h																R/W-0h															

Table 15-74. PWM3_CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM3_CMPA	R/W	0h	PWM3 Compare A register The value in the PWM3_CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM3 counter-compare module generates a 'time-base counter equal to CMP A' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM3A or the PWM3B output depending on the configuration of the PWM3_AQCTLA and PWM3_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM3_LOADAMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.2.61 PWM3_CMPAS Register (Offset = 282h) [Reset = 00000000h]

PWM3_CMPAS is shown in Figure 15-100 and described in Table 15-75.

Return to the Summary Table.

PWM3 counter compare A shadow register

Figure 15-100. PWM3_CMPAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM3_CMPAS															
R-0-0h																R/W-0h															

Table 15-75. PWM3_CMPAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM3_CMPAS	R/W	0h	PWM3 Compare A Shadow Register The value in the PWM3_CMPAS register is loaded into PWM3_CMPA register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.62 PWM3_CMPB Register (Offset = 284h) [Reset = 00000000h]

PWM3_CMPB is shown in Figure 15-101 and described in Table 15-76.

Return to the Summary Table.

PWM3 counter compare B register

Figure 15-101. PWM3_CMPB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM3_CMPB															
R-0-0h																R/W-0h															

Table 15-76. PWM3_CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM3_CMPB	R/W	0h	PWM3 Compare B register The value in the PWM3_CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM3 counter-compare module generates a 'time-base counter equal to CMP B' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM3A or the PWM3B output depending on the configuration of the PWM3_AQCTLA and PWM3_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM3_LOADBMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.2.63 PWM3_CMPBS Register (Offset = 286h) [Reset = 00000000h]

PWM3_CMPBS is shown in Figure 15-102 and described in Table 15-77.

Return to the Summary Table.

PWM3 counter compare B shadow register

Figure 15-102. PWM3_CMPBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM3_CMPBS															
R-0-0h																R/W-0h															

Table 15-77. PWM3_CMPBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM3_CMPBS	R/W	0h	PWM3 Compare B Shadow Register The value in the PWM3 CMPBS register is loaded into PWM3 CMPB register when shadow to active load occurs. Reset type: SYSRSn

15.12.2.64 PWM3_AQCTLA Register (Offset = 290h) [Reset = 00000000h]

PWM3_AQCTLA is shown in Figure 15-103 and described in Table 15-78.

Return to the Summary Table.

PWM3 action qualifier A register

Figure 15-103. PWM3_AQCTLA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-78. PWM3_AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM3_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM3_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM3_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM3_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-78. PWM3_AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.65 PWM3_AQCTLAS Register (Offset = 292h) [Reset = 00000000h]

PWM3_AQCTLAS is shown in Figure 15-104 and described in Table 15-79.

Return to the Summary Table.

PWM3 action qualifier A shadow register

Figure 15-104. PWM3_AQCTLAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-79. PWM3_AQCTLAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM3_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM3_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM3_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM3_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-79. PWM3_AQCTLAS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.66 PWM3_AQCTLB Register (Offset = 294h) [Reset = 00000000h]

PWM3_AQCTLB is shown in Figure 15-105 and described in Table 15-80.

Return to the Summary Table.

PWM3 action qualifier B register

Figure 15-105. PWM3_AQCTLB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-80. PWM3_AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM3_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM3_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM3_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM3_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-80. PWM3_AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.67 PWM3_AQCTLBS Register (Offset = 296h) [Reset = 00000000h]

PWM3_AQCTLBS is shown in Figure 15-106 and described in Table 15-81.

Return to the Summary Table.

PWM3 action qualifier B shadow register

Figure 15-106. PWM3_AQCTLBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-81. PWM3_AQCTLBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM3_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM3_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM3_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM3_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-81. PWM3_AQCTLBS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.2.68 PWM3_AQSFRC Register (Offset = 298h) [Reset = 00000000h]

PWM3_AQSFRC is shown in Figure 15-107 and described in Table 15-82.

Return to the Summary Table.

PWM3 action qualifier software force

Figure 15-107. PWM3_AQSFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	PWMB			RESERVED	PWMA		
R-0-0h	R/W-0h			R-0-0h	R/W-0h		

Table 15-82. PWM3_AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6-4	PWMB	R/W	0h	Action qualifier software force on PWMB 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output B 010: Forces a continuous high on output B 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM3_AQOTSFRC[PWMB] = '1'. 110: Set (high) when PWM3_AQOTSFRC[PWMB] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM3_AQOTSFRC[PWMB] = '1'. Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved
2-0	PWMA	R/W	0h	Action qualifier software force on PWMA 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output A 010: Forces a continuous high on output A 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM3_AQOTSFRC[PWMA] = '1'. 110: Set (high) when PWM3_AQOTSFRC[PWMA] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM3_AQOTSFRC[PWMA] = '1'. Reset type: SYSRSn

15.12.2.69 PWM3_AQOTSFRC Register (Offset = 29Ah) [Reset = 00000000h]

PWM3_AQOTSFRC is shown in Figure 15-108 and described in Table 15-83.

Return to the Summary Table.

PWM3 action qualifier one time software force

Figure 15-108. PWM3_AQOTSFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			PWMB	RESERVED			PWMA
R-0-0h			R-0/W1S-0h	R-0-0h			R-0/W1S-0h

Table 15-83. PWM3_AQOTSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4	PWMB	R-0/W1S	0h	Action qualifier one time software force on PWMB 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMB. 1: Initiates a single software forced event Reset type: SYSRSn
3-1	RESERVED	R-0	0h	Reserved
0	PWMA	R-0/W1S	0h	Action qualifier one time software force on PWMA 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMA. 1: Initiates a single software forced event Reset type: SYSRSn

15.12.3 MCPWM_2CH_REGS Registers

Table 15-84 lists the memory-mapped registers for the MCPWM_2CH_REGS registers. All register offset addresses not listed in Table 15-84 should be considered as reserved locations and the register contents should not be modified.

Table 15-84. MCPWM_2CH_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	REVISION	IP revision id register	
8h	TBCTL	Time base control register	
Ah	TBPRD	Time base period register	
Ch	TBPRDS	Time base period shadow register	
Eh	TBPHS	Time base phase offset register	
10h	TBSTS	Time base status register	
12h	TBSTSCLR	Time base status clear register	
14h	TBCTR	Time base counter register	
18h	CMPCTL	Counter compare control register	
20h	CMPC	Counter compare C register	
22h	CMPD	Counter compare D register	
24h	CMPCS	Counter compare C shadow register	
26h	CMPDS	Counter compare D shadow register	
28h	AQCTL	Action qualifier control register	
30h	SOCEN	Start of conversion enable	
32h	SOCSEL	Start of conversion selection	
34h	SOCPERIOD	Start of conversion period	
36h	SOCCNT	Start of conversion count	
38h	SOCFLAG	Start of conversion flag	
3Ah	SOCCLR	Start of conversion clear	
40h	ETSEL	Event trigger selection	
42h	ETPERIOD	Event trigger period	
44h	ETCNT	Event trigger count	
48h	INTEN	Interrupt enable	EALLOW
4Ah	INTFLAG	Interrupt flag	
4Ch	INTCLR	Interrupt clear	EALLOW
4Eh	INTFRC	Interrupt force	EALLOW
50h	TZSEL	Trip Zone selection	EALLOW
56h	TZCTL	Trip Zone control	EALLOW
58h	TZCBCOSTFLAG	Trip zone CBCOST flag	
5Ah	TZCBCOSTCLR	Trip zone CBCOST flag clear	EALLOW
60h	DBCTL	Dead band control register	
68h	DBFED	Dead band fall edge delay	
6Ah	DBRED	Dead band rise edge delay	
6Ch	DBFEDS	Dead band fall edge delay shadow register	
6Eh	DBREDS	Dead band rise edge delay shadow register	
78h	GLDCTL	Global load control register	EALLOW
7Ah	GLDOSHTCTL	Global load one shot control register	
7Ch	GLDOSHTSTS	Global load one shot status register	
80h	PWM1_CMPA	PWM1 counter compare A register	

Table 15-84. MCPWM_2CH_REGS Registers (continued)

Offset	Acronym	Register Name	Write Protection
82h	PWM1_CMPAS	PWM1 counter compare A shadow register	
84h	PWM1_CMPB	PWM1 counter compare B register	
86h	PWM1_CMPBS	PWM1 counter compare B shadow register	
90h	PWM1_AQCTLA	PWM1 action qualifier A register	
92h	PWM1_AQCTLAS	PWM1 action qualifier A shadow register	
94h	PWM1_AQCTLB	PWM1 action qualifier B register	
96h	PWM1_AQCTLBS	PWM1 action qualifier B shadow register	
98h	PWM1_AQSFRC	PWM1 action qualifier software force	
9Ah	PWM1_AQOTSFRC	PWM1 action qualifier one time software force	

Complex bit access types are encoded to fit into small table cells. Table 15-85 shows the codes that are used for access types in this section.

Table 15-85. MCPWM_2CH_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

15.12.3.1 REVISION Register (Offset = 0h) [Reset = 00000004h]

REVISION is shown in Figure 15-109 and described in Table 15-86.

Return to the Summary Table.

IP revision id register

Figure 15-109. REVISION Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				CAP_PRESENT	CMPCD_PRESENT	PWM3_PRESENT	PWM2_PRESENT
R-0-0h				R-0h	R-1h	R-0h	R-0h

Table 15-86. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	CAP_PRESENT	R	0h	This hardcoded field defines the presence of Capture mode feature. Reset type: SYSRSn
2	CMPCD_PRESENT	R	1h	This hardcoded field defines the presence of Compare C and D registers. Reset type: SYSRSn
1	PWM3_PRESENT	R	0h	This hardcoded field defines the presence of PWM3 channels. Reset type: SYSRSn
0	PWM2_PRESENT	R	0h	This hardcoded field defines the presence of PWM2 channels. Reset type: SYSRSn

15.12.3.2 TBCTL Register (Offset = 8h) [Reset = 00000002h]

TBCTL is shown in Figure 15-110 and described in Table 15-87.

Return to the Summary Table.

Time base control register

Figure 15-110. TBCTL Register

31	30	29	28	27	26	25	24
RESERVED		SYNCISEL					SYNCPERSEL
R-0-0h		R/W-0h					R/W-0h
23	22	21	20	19	18	17	16
SYNCPERSEL		FREE_SOFT		RESERVED	SYNCOSEL		
R/W-0h		R/W-0h		R-0-0h	R/W-0h		
15	14	13	12	11	10	9	8
SWSYNC	RESERVED			PHSDIR	PHSEN	RESERVED	PRDLD
R-0/W1S-0h	R-0-0h			R/W-0h	R/W-0h	R-0-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		CLKDIV				CTRMODE	
R-0-0h		R/W-0h				R/W-2h	

Table 15-87. TBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	Reserved
29-25	SYNCISEL	R/W	0h	These bits determines the source of SYNCIN signal. 0x0 : Disabled using SOC tieoff. 0x1-0x1F : Refer to PWM chapter of TRM. Reset type: SYSRSn
24-22	SYNCPERSEL	R/W	0h	Sync peripheral Select 000: Reserved (Disabled) 001: Reserved (Disabled) 010: CTR = PRD 011: CTR = 0 100: CTR = CMPC, Count direction Up 101: CTR = CMPC, Count direction Down 110: CTR = CMPD, Count direction Up 111: CTR = CMPD, Count direction Down Reset type: SYSRSn
21-20	FREE_SOFT	R/W	0h	Emulation Mode Bits. These bits select the behavior of the time-base counter during emulation events 00: Stop after the next time-base counter increment or decrement 01: Stop when counter completes a whole cycle: - Up-count mode: stop when the time-base counter = period (TBCTR = TBPRD) - Up-down-count mode: stop when the time-base counter = 0x00 (TBCTR = 0x00) 1x: Free run Reset type: SYSRSn
19	RESERVED	R-0	0h	Reserved

Table 15-87. TBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-16	SYNCOSEL	R/W	0h	Sync Output Select 000: SWFSYNC 001: CTR = zero: Time-base counter equal to zero (TBCTR = 0x00) 010: CTR = CMPC : Time-base counter equal to counter-compare C (TBCTR = CMPC) 011: CTR = CMPD : Time-base counter equal to counter-compare D (TBCTR = CMPD) 1xx: Disabled MCPWMxSYNCO sync signal Reset type: SYSRSn
15	SWSYNC	R-0/W1S	0h	Software Forced Sync Pulse 0: Writing a 0 has no effect and reads always return a 0. 1: Writing a 1 forces a one-time synchronization pulse to be generated. Reset type: SYSRSn
14-12	RESERVED	R-0	0h	Reserved
11	PHSDIR	R/W	0h	Phase Direction Bit. This bit is only used when the time-base counter is configured in the up-down-count mode. In the up-count mode, this bit is ignored. The bit indicates the direction of the time-base counter (TBCTR) after a sync event occurs and a new phase value is loaded from the phase (TBPHS) register. 0: Count down after the sync event. 1: Count up after the sync event. Reset type: SYSRSn
10	PHSEN	R/W	0h	Load Phase register to time-base counter(TBCNTR) 0: Do not load the time-base counter (TBCTR) from the time-base phase register (TBPHS). 1: Allow Counter to be loaded from the Phase register (TBPHS) when an MCPWMxSYNCI input signal occurs or a software-forced sync (SWSYNC). Reset type: SYSRSn
9	RESERVED	R-0	0h	Reserved
8	PRDLD	R/W	0h	Shadow to Active load of TBPRD register 0: Shadow to Active Load of TBPRD occurs when TBCTR = 0 1 : Disabled shadow to active load of TBPRD Reset type: SYSRSn
7-6	RESERVED	R-0	0h	Reserved
5-2	CLKDIV	R/W	0h	Time Base Clock Pre-Scale Bits These bits select the time base clock pre-scale value (TBCLK = MCPWMCLK/CLKDIV): 0000: /1 (default on reset) 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 1001: /512 1010: /1024 1011: /2048 1100: /4096 1101: /8192 1110: /16384 1111: /32768 Reset type: SYSRSn

Table 15-87. TBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	CTRMODE	R/W	2h	The time-base counter mode is normally configured once and not changed during normal operation. If you change the mode of the counter, the change will take effect at the next TBCLK edge and the current counter value shall increment or decrement from the value before the mode change. These bits set the time-base counter mode of operation as follows: 00: Up-count mode 01: Up-down count mode 1x: Freeze counter operation (default on reset) Reset type: SYSRSn

15.12.3.3 TBPRD Register (Offset = Ah) [Reset = 00000000h]

TBPRD is shown in Figure 15-111 and described in Table 15-88.

Return to the Summary Table.

Time base period register

Figure 15-111. TBPRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TBPRD															
R-0-0h																R/W-0h															

Table 15-88. TBPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBPRD	R/W	0h	Time Base Period Register These bits determine the period of the time-base counter. This sets the PWM frequency. Shadowing of this register is enabled and disabled by the TBCTL[PRDLD] bit. By default this register is shadowed. - If TBCTL[PRDLD] = 0, then the shadow is enabled. This register will also loaded from the shadow register when the time-base counter equals zero. - If TBCTL[PRDLD] = 1, then the shadow is disabled. Reset type: SYSRSn

15.12.3.4 TBPRDS Register (Offset = Ch) [Reset = 00000000h]

TBPRDS is shown in Figure 15-112 and described in Table 15-89.

Return to the Summary Table.

Time base period shadow register

Figure 15-112. TBPRDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TBPRDS															
R-0-0h																R/W-0h															

Table 15-89. TBPRDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBPRDS	R/W	0h	Time Base Period Shadow Register The value in the TBPRDS register is loaded into TBPRD register when shadow to active load occurs. Reset type: SYSRSn

15.12.3.5 TBPHS Register (Offset = Eh) [Reset = 00000000h]

TBPHS is shown in Figure 15-113 and described in Table 15-90.

Return to the Summary Table.

Time base phase offset register

Figure 15-113. TBPHS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TBPHS															
R-0-0h																R/W-0h															

Table 15-90. TBPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBPHS	R/W	0h	Phase Offset Register These bits set time-base counter phase of the PWM relative to the sync (MCPWMxSYNCI / SWFSYNC) - If TBCTL[PHSEN] = 0, then the sync event is ignored. - If TBCTL[PHSEN] = 1, then the time-base counter (TBCTR) will be loaded with the phase (TBPHS) when a sync event occurs. The sync event can be initiated by the input sync signal (MCPWMxSYNCI) or by a software forced sync. Reset type: SYSRSn

15.12.3.6 TBSTS Register (Offset = 10h) [Reset = 00000000h]

TBSTS is shown in Figure 15-114 and described in Table 15-91.

Return to the Summary Table.

Time base status register

Figure 15-114. TBSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						SYNCI	CTRDIR
R-0-0h						R-0h	R-0h

Table 15-91. TBSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	SYNCI	R	0h	Input Synchronization Latched Status Bit 0: No external sync event has occurred. 1: External sync event has occurred (MCPWMxSYNCI). Reset type: SYSRSn
0	CTRDIR	R	0h	Time Base Counter Direction Status Bit 0: Time-Base Counter is currently counting up. 1: Time-Base Counter is currently counting down. Note: This bit is only valid when the counter is not frozen. Reset type: SYSRSn

15.12.3.7 TBSTSCLR Register (Offset = 12h) [Reset = 00000000h]

TBSTSCLR is shown in Figure 15-115 and described in Table 15-92.

Return to the Summary Table.

Time base status clear register

Figure 15-115. TBSTSCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						SYNCI	RESERVED
R-0-0h						R-0/W1S-0h	R-0-0h

Table 15-92. TBSTSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	SYNCI	R-0/W1S	0h	Input Synchronization Latched Status Clear 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TBSTS[SYNCI] bit. Reset type: SYSRSn
0	RESERVED	R-0	0h	Reserved

15.12.3.8 TBCTR Register (Offset = 14h) [Reset = 00000000h]

TBCTR is shown in Figure 15-116 and described in Table 15-93.

Return to the Summary Table.

Time base counter register

Figure 15-116. TBCTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TBCTR															
R-0-0h																R/W-0h															

Table 15-93. TBCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBCTR	R/W	0h	Time Base Counter Register Reset type: SYSRSn

15.12.3.9 CMPCTL Register (Offset = 18h) [Reset = 00000000h]

CMPCTL is shown in Figure 15-117 and described in Table 15-94.

Return to the Summary Table.

Counter compare control register

Figure 15-117. CMPCTL Register

31	30	29	28	27	26	25	24
RESERVED				LOADDMODE		LOADCMODE	
R-0-0h				R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				PWM1_LOADBMODE		PWM1_LOADAMODE	
R-0-0h				R/W-0h		R/W-0h	

Table 15-94. CMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	Reserved
27-26	LOADDMODE	R/W	0h	Shadow to Active load of CMPD register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter CCM_CD_PRESENT = 1 Reset type: SYSRSn
25-24	LOADCMODE	R/W	0h	Shadow to Active load of CMPC register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter CCM_CD_PRESENT = 1 Reset type: SYSRSn
23-4	RESERVED	R-0	0h	Reserved
3-2	PWM1_LOADBMODE	R/W	0h	Shadow to Active load of PWM1_CMPB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn

Table 15-94. CMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	PWM1_LOADAMODE	R/W	0h	Shadow to Active load of PWM1_CMPA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn

15.12.3.10 CMPC Register (Offset = 20h) [Reset = 00000000h]

CMPC is shown in Figure 15-118 and described in Table 15-95.

Return to the Summary Table.

Counter compare C register

Figure 15-118. CMPC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CMPC															
R-0-0h																R/W-0h															

Table 15-95. CMPC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPC	R/W	0h	Compare C register The value in the CMPC register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare C' event. Shadowing of this register is enabled and disabled by the CMPCTL[LOADCMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.3.11 CMPD Register (Offset = 22h) [Reset = 00000000h]

CMPD is shown in Figure 15-119 and described in Table 15-96.

Return to the Summary Table.

Counter compare D register

Figure 15-119. CMPD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CMPD															
R-0-0h																R/W-0h															

Table 15-96. CMPD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPD	R/W	0h	Compare D register The value in the CMPD register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare D' event. Shadowing of this register is enabled and disabled by the CMPCTL[LOADDMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.3.12 CMPCS Register (Offset = 24h) [Reset = 00000000h]

CMPCS is shown in Figure 15-120 and described in Table 15-97.

Return to the Summary Table.

Counter compare C shadow register

Figure 15-120. CMPCS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CMPCS															
R-0-0h																R/W-0h															

Table 15-97. CMPCS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPCS	R/W	0h	Compare C Shadow Register The value in the CMPCS register is loaded into CMPC register when shadow to active load occurs. Reset type: SYSRSn

15.12.3.13 CMPDS Register (Offset = 26h) [Reset = 00000000h]

CMPDS is shown in Figure 15-121 and described in Table 15-98.

Return to the Summary Table.

Counter compare D shadow register

Figure 15-121. CMPDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CMPDS															
R-0-0h																R/W-0h															

Table 15-98. CMPDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPDS	R/W	0h	Compare D Shadow Register The value in the CMPDS register is loaded into CMPD register when shadow to active load occurs. Reset type: SYSRSn

15.12.3.14 AQCTL Register (Offset = 28h) [Reset = 00000000h]

AQCTL is shown in Figure 15-122 and described in Table 15-99.

Return to the Summary Table.

Action qualifier control register

Figure 15-122. AQCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				PWM1_LDAQBMODE		PWM1_LDAQAMODE	
R-0-0h				R/W-0h		R/W-0h	

Table 15-99. AQCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3-2	PWM1_LDAQBMODE	R/W	0h	Shadow to Active load of PWM1_AQCTLB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
1-0	PWM1_LDAQAMODE	R/W	0h	Shadow to Active load of PWM1_AQCTLA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn

15.12.3.15 SOCEN Register (Offset = 30h) [Reset = 00000000h]

SOCEN is shown in Figure 15-123 and described in Table 15-100.

Return to the Summary Table.

Start of conversion enable

Figure 15-123. SOCEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	SOCB_ENABLE	SOCA_ENABLE
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-100. SOCEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	SOCB_ENABLE	R/W	0h	SOCB Selection Enable 0 - SOCB disable 1- SOCB enable Reset type: SYSRSn
0	SOCA_ENABLE	R/W	0h	SOCA Selection Enable 0 - SOCA disable 1- SOCA enable Reset type: SYSRSn

15.12.3.16 SOCSEL Register (Offset = 32h) [Reset = 00000000h]

SOCSEL is shown in Figure 15-124 and described in Table 15-101.

Return to the Summary Table.

Start of conversion selection

Figure 15-124. SOCSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED			RESERVED					RESERVED			RESERVED				
R-0-0h			R/W-0h					R-0-0h			R/W-0h				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			SOCB_SEL					RESERVED			SOCA_SEL				
R-0-0h			R/W-0h					R-0-0h			R/W-0h				

Table 15-101. SOCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R-0	0h	Reserved
28-24	RESERVED	R/W	0h	Reserved
23-21	RESERVED	R-0	0h	Reserved
20-16	RESERVED	R/W	0h	Reserved
15-13	RESERVED	R-0	0h	Reserved

Table 15-101. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-8	SOCB_SEL	R/W	0h	SOCB Selection Options These bits determine when SOCB pulse will be generated. 00000: Reserved (SOC B disabled) 00001: Reserved (SOC B disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC B disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC B disabled) Reset type: SYSRSn
7-5	RESERVED	R-0	0h	Reserved

Table 15-101. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	SOCA_SEL	R/W	0h	SOCA Selection Options These bits determine when SOCA pulse will be generated. 00000: Reserved (SOC A disabled) 00001: Reserved (SOC A disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC A disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC A disabled) Reset type: SYSRSn

15.12.3.17 SOCPERIOD Register (Offset = 34h) [Reset = 00000000h]

SOCPERIOD is shown in Figure 15-125 and described in Table 15-102.

Return to the Summary Table.

Start of conversion period

Figure 15-125. SOCPERIOD Register

31	30	29	28	27	26	25	24
RESERVED					RESERVED		
R-0-0h					R/W-0h		
23	22	21	20	19	18	17	16
RESERVED					RESERVED		
R-0-0h					R/W-0h		
15	14	13	12	11	10	9	8
RESERVED					SOCB_PERIOD		
R-0-0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					SOCA_PERIOD		
R-0-0h					R/W-0h		

Table 15-102. SOCPERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	Reserved
26-24	RESERVED	R/W	0h	Reserved
23-19	RESERVED	R-0	0h	Reserved
18-16	RESERVED	R/W	0h	Reserved
15-11	RESERVED	R-0	0h	Reserved
10-8	SOCB_PERIOD	R/W	0h	SOCB Period These bits determine how many selected SOCSEL[SOCB_SEL] events need to occur before an SOCB pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCB_ENABLE] = 1). The SOCB pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCB] = 1). Once the SOCB pulse is generated, the SOCCNT[SOCB_CNT] bits will automatically be cleared. Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	SOCA_PERIOD	R/W	0h	SOCA Period These bits determine how many selected SOCSEL[SOCA_SEL] events need to occur before an SOCA pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCA_ENABLE] = 1). The SOCA pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCA] = 1). Once the SOCA pulse is generated, the SOCCNT[SOCA_CNT] bits will automatically be cleared. Reset type: SYSRSn

15.12.3.18 SOCCNT Register (Offset = 36h) [Reset = 00000000h]

SOCCNT is shown in Figure 15-126 and described in Table 15-103.

Return to the Summary Table.

Start of conversion count

Figure 15-126. SOCCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					RESERVED			RESERVED					RESERVED		
R-0-0h					R-0h			R-0-0h					R-0h		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					SOCB_CNT			RESERVED					SOCA_CNT		
R-0-0h					R-0h			R-0-0h					R-0h		

Table 15-103. SOCCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	Reserved
26-24	RESERVED	R	0h	Reserved
23-19	RESERVED	R-0	0h	Reserved
18-16	RESERVED	R	0h	Reserved
15-11	RESERVED	R-0	0h	Reserved
10-8	SOCB_CNT	R	0h	SOC B Counter Register These bits indicate how many selected SOCSEL[SOCB_SEL] events have occurred. These bits are automatically cleared when a SOCB pulse is generated. Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	SOCA_CNT	R	0h	SOC A Counter Register These bits indicate how many selected SOCSEL[SOCA_SEL] events have occurred. These bits are automatically cleared when a SOCA pulse is generated. Reset type: SYSRSn

15.12.3.19 SOCFLAG Register (Offset = 38h) [Reset = 00000000h]

SOCFLAG is shown in Figure 15-127 and described in Table 15-104.

Return to the Summary Table.

Start of conversion flag

Figure 15-127. SOCFLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	SOCB	SOCA
R-0-0h				R-0h	R-0h	R-0h	R-0h

Table 15-104. SOCFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	SOCB	R	0h	Latched SOC B Status Flag SOCB output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCB. The SOCB output will continue to be generated even if the flag bit is set. Reset type: SYSRSn
0	SOCA	R	0h	Latched SOC A Status Flag SOCA output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCA. The SOCA output will continue to be generated even if the flag bit is set. Reset type: SYSRSn

15.12.3.20 SOCCLR Register (Offset = 3Ah) [Reset = 00000000h]

SOCCLR is shown in Figure 15-128 and described in Table 15-105.

Return to the Summary Table.

Start of conversion clear

Figure 15-128. SOCCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	SOCB	SOCA
R-0-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 15-105. SOCCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	RESERVED	R-0/W1S	0h	Reserved
2	RESERVED	R-0/W1S	0h	Reserved
1	SOCB	R-0/W1S	0h	Clear SOC B Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCB] bit. Reset type: SYSRSn
0	SOCA	R-0/W1S	0h	Clear SOC A Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCA] bit. Reset type: SYSRSn

15.12.3.21 ETSEL Register (Offset = 40h) [Reset = 00000000h]

ETSEL is shown in Figure 15-129 and described in Table 15-106.

Return to the Summary Table.

Event trigger selection

Figure 15-129. ETSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ET2_SEL					RESERVED			ET1_SEL				
R-0-0h			R/W-0h					R-0-0h			R/W-0h				

Table 15-106. ETSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	Reserved

Table 15-106. ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-8	ET2_SEL	R/W	0h	Event trigger2 Selection Options These bits determine when event trigger pulse will be generated. 00000: Reserved (ET2 Disabled) 00001: Reserved (ET2 Disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (ET2 Disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (ET2 Disabled) Reset type: SYSRSn
7-5	RESERVED	R-0	0h	Reserved

Table 15-106. ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	ET1_SEL	R/W	0h	Event trigger1 Selection Options These bits determine when event trigger pulse will be generated. 00000: Reserved (ET1 Disabled) 00001: Reserved (ET1 Disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (ET1 Disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (ET1 Disabled) Reset type: SYSRSn

15.12.3.22 ETPERIOD Register (Offset = 42h) [Reset = 00000000h]

ETPERIOD is shown in Figure 15-130 and described in Table 15-107.

Return to the Summary Table.

Event trigger period

Figure 15-130. ETPERIOD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED					ET2_PERIOD		
R-0-0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					ET1_PERIOD		
R-0-0h					R/W-0h		

Table 15-107. ETPERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	Reserved
10-8	ET2_PERIOD	R/W	0h	These bits determine how many selected ETSEL[ET2_SEL] events need to occur before an interrupt is generated. If the interrupt status flag is set from a previous interrupt (INTFLG[ET2] = 1) then no interrupt will be generated until the flag is cleared via the INTCLR[ET2] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETCNT[ET2_CNT] bits will automatically be cleared. Writing a PERIOD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a PERIOD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero PERIOD value is written, the counter is incremented. Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	ET1_PERIOD	R/W	0h	These bits determine how many selected ETSEL[ET1_SEL] events need to occur before an interrupt is generated. If the interrupt status flag is set from a previous interrupt (INTFLG[ET1] = 1) then no interrupt will be generated until the flag is cleared via the INTCLR[ET1] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETCNT[ET1_CNT] bits will automatically be cleared. Writing a PERIOD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a PERIOD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero PERIOD value is written, the counter is incremented. Reset type: SYSRSn

15.12.3.23 ETCNT Register (Offset = 44h) [Reset = 00000000h]

ETCNT is shown in Figure 15-131 and described in Table 15-108.

Return to the Summary Table.

Event trigger count

Figure 15-131. ETCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					ET2_CNT			RESERVED					ET1_CNT		
R-0-0h					R-0h			R-0-0h					R-0h		

Table 15-108. ETCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	Reserved
10-8	ET2_CNT	R	0h	Event trigger2 Counter Register These bits indicate how many selected ET_SEL[ET2_SEL] events have occurred. These bits are automatically cleared once INTFLAG.ET2 is generated (irrespective of INTEN.ET2 configuration). Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	ET1_CNT	R	0h	Event trigger1 Counter Register These bits indicate how many selected ET_SEL[ET1_SEL] events have occurred. These bits are automatically cleared once INTFLAG.ET1 is generated (irrespective of INTEN.ET1 configuration). Reset type: SYSRSn

15.12.3.24 INTEN Register (Offset = 48h) [Reset = 00000000h]

INTEN is shown in Figure 15-132 and described in Table 15-109.

Return to the Summary Table.

Interrupt enable

Figure 15-132. INTEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	RESERVED
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 15-109. INTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R/W	0h	Counter Overflow Interrupt Enable 1 : Enables CNT_OVF interrupt 0 : Disable CNT_OVF interrupt Reset type: SYSRSn
4	ET2	R/W	0h	Event Trigger 2 Interrupt Enable 1 : Enables ET2 interrupt 0 : Disable ET2 interrupt Reset type: SYSRSn
3	ET1	R/W	0h	Event Trigger 1 Interrupt Enable 1 : Enables ET1 interrupt 0 : Disable ET1 interrupt Reset type: SYSRSn
2	OST	R/W	0h	Trip-zone One-Shot Interrupt Enable 1 : Enables OST interrupt 0 : Disable OST interrupt Reset type: SYSRSn
1	CBC	R/W	0h	Trip-zone Cycle-by-Cycle Interrupt Enable 1 : Enables CBC interrupt 0 : Disable CBC interrupt Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

15.12.3.25 INTFLAG Register (Offset = 4Ah) [Reset = 00000000h]

INTFLAG is shown in Figure 15-133 and described in Table 15-110.

Return to the Summary Table.

Interrupt flag

Figure 15-133. INTFLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	INT
R-0-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-110. INTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R	0h	Latched Status Flag for A Counter Overflow Event 1 : CNT_OVF flag is set 0 : CNT_OVF flag is not set Reset type: SYSRSn
4	ET2	R	0h	Latched Status Flag for A Event Trigger 2 Event 1: ET2 flag is set 0: ET2 flag is not set Reset type: SYSRSn
3	ET1	R	0h	Latched Status Flag for A Event Trigger 1 Event 1: ET1 flag is set 0: ET1 flag is not set Reset type: SYSRSn
2	OST	R	0h	Latched Status Flag for A One-Shot Trip Event 1: OST flag is set 0: OST flag is not set Reset type: SYSRSn
1	CBC	R	0h	Latched Status Flag for Cycle-By-Cycle Trip Event 1: CBC flag is set 0: CBC flag is not set Reset type: SYSRSn
0	INT	R	0h	Global Interrupt Status Flag 1: Global flag is set 0: Global flag is not set Reset type: SYSRSn

15.12.3.26 INTCLR Register (Offset = 4Ch) [Reset = 00000000h]

INTCLR is shown in Figure 15-134 and described in Table 15-111.

Return to the Summary Table.

Interrupt clear

Figure 15-134. INTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	INT
R-0-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 15-111. INTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R-0/W1S	0h	Clear Counter Overflow Flag Writing '1' will clear INTFLAG[CNT_OVF] register. Reset type: SYSRSn
4	ET2	R-0/W1S	0h	Clear Event Trigger 2 Flag Writing '1' will clear INTFLAG[ET2] register. Reset type: SYSRSn
3	ET1	R-0/W1S	0h	Clear Event Trigger 1 Flag Writing '1' will clear INTFLAG[ET1] register. Reset type: SYSRSn
2	OST	R-0/W1S	0h	Clear One-Shot Trip Latch Writing '1' will clear INTFLAG[OST] register. Reset type: SYSRSn
1	CBC	R-0/W1S	0h	Clear Cycle-by-Cycle Trip Latch Writing '1' will clear INTFLAG[CBC] register. Reset type: SYSRSn
0	INT	R-0/W1S	0h	Clear Global Interrupt Flag Writing '1' will clear INTFLAG[INT] register. Reset type: SYSRSn

15.12.3.27 INTFRC Register (Offset = 4Eh) [Reset = 00000000h]

INTFRC is shown in Figure 15-135 and described in Table 15-112.

Return to the Summary Table.

Interrupt force

Figure 15-135. INTFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	RESERVED
R-0-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 15-112. INTFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R-0/W1S	0h	Force Counter Overflow Interrupt Writing '1' will set INTFLAG[CNT_OVF] register. Reset type: SYSRSn
4	ET2	R-0/W1S	0h	Force Event Trigger 2 Interrupt Writing '1' will set INTFLAG[ET2] register. Reset type: SYSRSn
3	ET1	R-0/W1S	0h	Force Event Trigger 1 Interrupt Writing '1' will set INTFLAG[ET1] register. Reset type: SYSRSn
2	OST	R-0/W1S	0h	Force One-Shot Trip Interrupt Writing '1' will set INTFLAG[OST] register. Reset type: SYSRSn
1	CBC	R-0/W1S	0h	Force Cycle-by-Cycle Trip Interrupt Writing '1' will set INTFLAG[CBC] register. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

15.12.3.28 TZSEL Register (Offset = 50h) [Reset = 00000000h]

TZSEL is shown in Figure 15-136 and described in Table 15-113.

Return to the Summary Table.

Trip Zone selection

Figure 15-136. TZSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
OST8	OST7	OST6	OST5	OST4	OST3	OST2	OST1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CBC8	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-113. TZSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	OST8	R/W	0h	Select Trip-zone 8 (TZ8) for OST generation 0: Disable TZ8 as a OST trip source for this MCPWM module 1: Enable TZ8 as a OST trip source for this MCPWM module Reset type: SYSRSn
22	OST7	R/W	0h	Select Trip-zone 7 (TZ7) for OST generation 0: Disable TZ7 as a OST trip source for this MCPWM module 1: Enable TZ7 as a OST trip source for this MCPWM module Reset type: SYSRSn
21	OST6	R/W	0h	Select Trip-zone 6 (TZ6) for OST generation 0: Disable TZ6 as a OST trip source for this MCPWM module 1: Enable TZ6 as a OST trip source for this MCPWM module Reset type: SYSRSn
20	OST5	R/W	0h	Select Trip-zone 5 (TZ5) for OST generation 0: Disable TZ5 as a OST trip source for this MCPWM module 1: Enable TZ5 as a OST trip source for this MCPWM module Reset type: SYSRSn
19	OST4	R/W	0h	Select Trip-zone 4 (TZ4) for OST generation 0: Disable TZ4 as a OST trip source for this MCPWM module 1: Enable TZ4 as a OST trip source for this MCPWM module Reset type: SYSRSn
18	OST3	R/W	0h	Select Trip-zone 3 (TZ3) for OST generation 0: Disable TZ3 as a OST trip source for this MCPWM module 1: Enable TZ3 as a OST trip source for this MCPWM module Reset type: SYSRSn
17	OST2	R/W	0h	Select Trip-zone 2 (TZ2) for OST generation 0: Disable TZ2 as a OST trip source for this MCPWM module 1: Enable TZ2 as a OST trip source for this MCPWM module Reset type: SYSRSn

Table 15-113. TZSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	OST1	R/W	0h	Select Trip-zone 1 (TZ1) for OST generation 0: Disable TZ1 as a OST trip source for this MCPWM module 1: Enable TZ1 as a OST trip source for this MCPWM module Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7	CBC8	R/W	0h	Select Trip-zone 8 (TZ8) for CBC generation 0: Disable TZ8 as a CBC trip source for this MCPWM module 1: Enable TZ8 as a CBC trip source for this MCPWM module Reset type: SYSRSn
6	CBC7	R/W	0h	Select Trip-zone 7 (TZ7) for CBC generation 0: Disable TZ7 as a CBC trip source for this MCPWM module 1: Enable TZ7 as a CBC trip source for this MCPWM module Reset type: SYSRSn
5	CBC6	R/W	0h	Select Trip-zone 6 (TZ6) for CBC generation 0: Disable TZ6 as a CBC trip source for this MCPWM module 1: Enable TZ6 as a CBC trip source for this MCPWM module Reset type: SYSRSn
4	CBC5	R/W	0h	Select Trip-zone 5 (TZ5) for CBC generation 0: Disable TZ5 as a CBC trip source for this MCPWM module 1: Enable TZ5 as a CBC trip source for this MCPWM module Reset type: SYSRSn
3	CBC4	R/W	0h	Select Trip-zone 4 (TZ4) for CBC generation 0: Disable TZ4 as a CBC trip source for this MCPWM module 1: Enable TZ4 as a CBC trip source for this MCPWM module Reset type: SYSRSn
2	CBC3	R/W	0h	Select Trip-zone 3 (TZ3) for CBC generation 0: Disable TZ3 as a CBC trip source for this MCPWM module 1: Enable TZ3 as a CBC trip source for this MCPWM module Reset type: SYSRSn
1	CBC2	R/W	0h	Select Trip-zone 2 (TZ2) for CBC generation 0: Disable TZ2 as a CBC trip source for this MCPWM module 1: Enable TZ2 as a CBC trip source for this MCPWM module Reset type: SYSRSn
0	CBC1	R/W	0h	Select Trip-zone 1 (TZ1) for CBC generation 0: Disable TZ1 as a CBC trip source for this MCPWM module 1: Enable TZ1 as a CBC trip source for this MCPWM module Reset type: SYSRSn

15.12.3.29 TZCTL Register (Offset = 56h) [Reset = 00000010h]

TZCTL is shown in Figure 15-137 and described in Table 15-114.

Return to the Summary Table.

Trip Zone control

Figure 15-137. TZCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CBCPULSE		TZB		TZA	
R-0-0h		R/W-1h		R/W-0h		R/W-0h	

Table 15-114. TZCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-4	CBCPULSE	R/W	1h	Clear Pulse for Cycle-By-Cycle (CBC) Trip Latch This bit field determines which pulse clears the CBC trip latch. 00: CBC trip latch is not cleared 01: CTR = ZERO pulse clears CBC trip latch 10: CTR = PRD pulse clears CBC trip latch 11: CTR = ZERO or CTR = PRD pulse clears CBC trip latch Reset type: SYSRSn
3-2	TZB	R/W	0h	Trip action on PWMxB 00: High-impedance (PWMxB = High-impedance state) 01: Force PWMxB to a high state 10: Force PWMxB to a low state 11: Do nothing, no action is taken on PWMxB. Reset type: SYSRSn
1-0	TZA	R/W	0h	Trip action on PWMxA 00: High-impedance (PWMxA = High-impedance state) 01: Force PWMxA to a high state 10: Force PWMxA to a low state 11: Do nothing, no action is taken on PWMxA. Reset type: SYSRSn

15.12.3.30 TZCBCOSTFLAG Register (Offset = 58h) [Reset = 00000000h]

TZCBCOSTFLAG is shown in Figure 15-138 and described in Table 15-115.

Return to the Summary Table.

Trip zone CBCOST flag

Figure 15-138. TZCBCOSTFLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
OST8	OST7	OST6	OST5	OST4	OST3	OST2	OST1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CBC8	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-115. TZCBCOSTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	OST8	R	0h	Latched Status Flag for OST TZ8 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ8. 1: Reading a 1 indicates a OST trip has occured by TZ8. Reset type: SYSRSn
22	OST7	R	0h	Latched Status Flag for OST TZ7 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ7. 1: Reading a 1 indicates a OST trip has occured by TZ7. Reset type: SYSRSn
21	OST6	R	0h	Latched Status Flag for OST TZ6 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ6. 1: Reading a 1 indicates a OST trip has occured by TZ6. Reset type: SYSRSn
20	OST5	R	0h	Latched Status Flag for OST TZ5 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ5. 1: Reading a 1 indicates a OST trip has occured by TZ5. Reset type: SYSRSn
19	OST4	R	0h	Latched Status Flag for OST TZ4 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ4. 1: Reading a 1 indicates a OST trip has occured by TZ4. Reset type: SYSRSn
18	OST3	R	0h	Latched Status Flag for OST TZ3 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ3. 1: Reading a 1 indicates a OST trip has occured by TZ3. Reset type: SYSRSn
17	OST2	R	0h	Latched Status Flag for OST TZ2 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ2. 1: Reading a 1 indicates a OST trip has occured by TZ2. Reset type: SYSRSn

Table 15-115. TZCBCOSTFLAG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	OST1	R	0h	Latched Status Flag for OST TZ1 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ1. 1: Reading a 1 indicates a OST trip has occured by TZ1. Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7	CBC8	R	0h	Latched Status Flag for CBC TZ8 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ8. 1: Reading a 1 indicates a CBC trip has occured by TZ8. Reset type: SYSRSn
6	CBC7	R	0h	Latched Status Flag for CBC TZ7 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ7. 1: Reading a 1 indicates a CBC trip has occured by TZ7. Reset type: SYSRSn
5	CBC6	R	0h	Latched Status Flag for CBC TZ6 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ6. 1: Reading a 1 indicates a CBC trip has occured by TZ6. Reset type: SYSRSn
4	CBC5	R	0h	Latched Status Flag for CBC TZ5 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ5. 1: Reading a 1 indicates a CBC trip has occured by TZ5. Reset type: SYSRSn
3	CBC4	R	0h	Latched Status Flag for CBC TZ4 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ4. 1: Reading a 1 indicates a CBC trip has occured by TZ4. Reset type: SYSRSn
2	CBC3	R	0h	Latched Status Flag for CBC TZ3 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ3. 1: Reading a 1 indicates a CBC trip has occured by TZ3. Reset type: SYSRSn
1	CBC2	R	0h	Latched Status Flag for CBC TZ2 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ2. 1: Reading a 1 indicates a CBC trip has occured by TZ2. Reset type: SYSRSn
0	CBC1	R	0h	Latched Status Flag for CBC TZ1 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ1. 1: Reading a 1 indicates a CBC trip has occured by TZ1. Reset type: SYSRSn

15.12.3.31 TZCBCOSTCLR Register (Offset = 5Ah) [Reset = 00000000h]

TZCBCOSTCLR is shown in Figure 15-139 and described in Table 15-116.

Return to the Summary Table.

Trip zone CBCOST flag clear

Figure 15-139. TZCBCOSTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
OST8	OST7	OST6	OST5	OST4	OST3	OST2	OST1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CBC8	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 15-116. TZCBCOSTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	OST8	R-0/W1S	0h	Clear OST TZ8 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST8] bit. Reset type: SYSRSn
22	OST7	R-0/W1S	0h	Clear OST TZ7 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST7] bit. Reset type: SYSRSn
21	OST6	R-0/W1S	0h	Clear OST TZ6 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST6] bit. Reset type: SYSRSn
20	OST5	R-0/W1S	0h	Clear OST TZ5 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST5] bit. Reset type: SYSRSn
19	OST4	R-0/W1S	0h	Clear OST TZ4 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST4] bit. Reset type: SYSRSn
18	OST3	R-0/W1S	0h	Clear OST TZ3 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST3] bit. Reset type: SYSRSn
17	OST2	R-0/W1S	0h	Clear OST TZ2 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST2] bit. Reset type: SYSRSn

Table 15-116. TZCBCOSTCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	OST1	R-0/W1S	0h	Clear OST TZ1 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST1] bit. Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7	CBC8	R-0/W1S	0h	Clear CBC TZ8 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC8] bit. Reset type: SYSRSn
6	CBC7	R-0/W1S	0h	Clear CBC TZ7 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC7] bit. Reset type: SYSRSn
5	CBC6	R-0/W1S	0h	Clear CBC TZ6 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC6] bit. Reset type: SYSRSn
4	CBC5	R-0/W1S	0h	Clear CBC TZ5 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC5] bit. Reset type: SYSRSn
3	CBC4	R-0/W1S	0h	Clear CBC TZ4 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC4] bit. Reset type: SYSRSn
2	CBC3	R-0/W1S	0h	Clear CBC TZ3 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC3] bit. Reset type: SYSRSn
1	CBC2	R-0/W1S	0h	Clear CBC TZ2 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC2] bit. Reset type: SYSRSn
0	CBC1	R-0/W1S	0h	Clear CBC TZ1 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC1] bit. Reset type: SYSRSn

15.12.3.32 DBCTL Register (Offset = 60h) [Reset = 00000000h]

DBCTL is shown in Figure 15-140 and described in Table 15-117.

Return to the Summary Table.

Dead band control register

Figure 15-140. DBCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED				LOADREDMODE		LOADFEDMODE	
R-0-0h				R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED							DEDB_MODE
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
OUTSWAP		IN_MODE		POLSEL		OUT_MODE	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-117. DBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	Reserved
19-18	LOADREDMODE	R/W	0h	Shadow to Active load of DBRED register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
17-16	LOADFEDMODE	R/W	0h	Shadow to Active load of DBFED register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
15-9	RESERVED	R-0	0h	Reserved
8	DEDB_MODE	R/W	0h	Dead Band Dual-Edge B Mode Control (S8 switch) 0: Rising edge delay applied to InA/InB as selected by S4 switch (IN-MODE bits) on A signal path only. Falling edge delay applied to InA/InB as selected by S5 switch (INMODE bits) on B signal path only. 1: Rising edge delay and falling edge delay applied to source selected by S4 switch (INMODE bits) and output to B signal path only. Note: When this bit is set to 1, user should always either set OUT_MODE bits such that Apath = InA OR OUTSWAP bits such that OutA=Bpath otherwise, OutA will be invalid. Reset type: SYSRSn

Table 15-117. DBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	OUTSWAP	R/W	0h	Dead Band Output Swap Control Bit 7 controls the S6 switch and bit 6 controls the S7 switch. 00: OutA and OutB signals are as defined by OUT-MODE bits. 01: OutA = A-path as defined by OUT-MODE bits. OutB = A-path as defined by OUT-MODE bits (rising edge delay or delay-bypassed A signal path). 10: OutA = B-path as defined by OUT-MODE bits (falling edge delay or delay-bypassed B signal path). OutB = B-path as defined by OUT-MODE bits. 11: OutA = B-path as defined by OUT-MODE bits (falling edge delay or delay-bypassed B signal path). OutB = A-path as defined by OUT-MODE bits (rising edge delay or delay-bypassed A signal path). Reset type: SYSRSn
5-4	IN_MODE	R/W	0h	Dead-Band Input Mode Control Bit 5 controls the S5 switch and bit 4 controls the S4 switch shown. This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms the default is PWMxA In is the source for both falling and rising-edge delays. 00: PWMxA In (from the action-qualifier) is the source for both falling-edge and rising-edge delay. 01: PWMxB In (from the action-qualifier) is the source for rising-edge delayed signal. PWMxA In (from the action-qualifier) is the source for falling-edge delayed signal. 10: PWMxA In (from the action-qualifier) is the source for rising-edge delayed signal. PWMxB In (from the action-qualifier) is the source for falling-edge delayed signal. 11: PWMxB In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal. Reset type: SYSRSn
3-2	POLSEL	R/W	0h	Polarity Select Control Bit 3 controls the S3 switch and bit 2 controls the S2 switch. This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule. The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that DBCTL[OUT_MODE] = 1,1 and DBCTL[IN_MODE] = 0x0. Other enhanced modes are also possible, but not regarded as typical usage modes. 00: Active high (AH) mode. Neither PWMxA nor PWMxB is inverted (default). 01: Active low complementary (ALC) mode. PWMxA is inverted. 10: Active high complementary (AHC). PWMxB is inverted. 11: Active low (AL) mode. Both PWMxA and PWMxB are inverted. Reset type: SYSRSn
1-0	OUT_MODE	R/W	0h	Dead-Band Output Mode Control Bit 1 controls the S1 switch and bit 0 controls the S0 switch. 00: DBM is fully disabled or by-passed. In this mode the POLSEL and IN-MODE bits have no effect. 01: Apath = InA (delay is by-passed for A signal path) Bpath = FED (Falling Edge Delay in B signal path) 10: Apath = RED (Rising Edge Delay in A signal path) Bpath = InB (delay is by-passed for B signal path) 11: DBM is fully enabled (i.e. both RED and FED active) Reset type: SYSRSn

15.12.3.33 DBFED Register (Offset = 68h) [Reset = 00000000h]

DBFED is shown in Figure 15-141 and described in Table 15-118.

Return to the Summary Table.

Dead band fall edge delay

Figure 15-141. DBFED Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		DBFED													
R-0-0h																		R/W-0h													

Table 15-118. DBFED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBFED	R/W	0h	Falling Edge Delay Count 14-bit counter Reset type: SYSRSn

15.12.3.34 DBRED Register (Offset = 6Ah) [Reset = 00000000h]

DBRED is shown in Figure 15-142 and described in Table 15-119.

Return to the Summary Table.

Dead band rise edge delay

Figure 15-142. DBRED Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		DBRED													
R-0-0h																		R/W-0h													

Table 15-119. DBRED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBRED	R/W	0h	Rising Edge Delay Count 14-bit counter Reset type: SYSRSn

15.12.3.35 DBFEDS Register (Offset = 6Ch) [Reset = 00000000h]

DBFEDS is shown in Figure 15-143 and described in Table 15-120.

Return to the Summary Table.

Dead band fall edge delay shadow register

Figure 15-143. DBFEDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		DBFEDS													
R-0-0h																		R/W-0h													

Table 15-120. DBFEDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBFEDS	R/W	0h	DBFED Shadow Register The value in the DBFEDS register is loaded into DBFED register when shadow to active load occurs. Reset type: SYSRSn

15.12.3.36 DBREDS Register (Offset = 6Eh) [Reset = 00000000h]

DBREDS is shown in Figure 15-144 and described in Table 15-121.

Return to the Summary Table.

Dead band rise edge delay shadow register

Figure 15-144. DBREDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		DBREDS													
R-0-0h																		R/W-0h													

Table 15-121. DBREDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBREDS	R/W	0h	DBRED Shadow Register The value in the DBREDS register is loaded into DBRED register when shadow to active load occurs. Reset type: SYSRSn

15.12.3.37 GLDCTL Register (Offset = 78h) [Reset = 00000000h]

GLDCTL is shown in Figure 15-145 and described in Table 15-122.

Return to the Summary Table.

Global load control register

Figure 15-145. GLDCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		GLDMODE		RESERVED		OSHTMODE	GLD
R-0-0h		R/W-0h		R-0-0h		R/W-0h	R/W-0h

Table 15-122. GLDCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-4	GLDMODE	R/W	0h	Select global load event 00: CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: CTR = Zero or CTR = PRD 11: GLDOSHTCTL[GFRCLD] - Softwrae load Reset type: SYSRSn
3-2	RESERVED	R-0	0h	Reserved
1	OSHTMODE	R/W	0h	Global load one-shot enable 0: Disable global load one-shot 1: Enable global load one-shot Reset type: SYSRSn
0	GLD	R/W	0h	Global load enable 0: Disable global load 1: Enable global load Reset type: SYSRSn

15.12.3.38 GLDOSHTCTL Register (Offset = 7Ah) [Reset = 00000000h]

GLDOSHTCTL is shown in Figure 15-146 and described in Table 15-123.

Return to the Summary Table.

Global load one shot control register

Figure 15-146. GLDOSHTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					GFRCLD	OSHTCLR	OSHTLD
R-0-0h					R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 15-123. GLDOSHTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	GFRCLD	R-0/W1S	0h	Force Load Event in One Shot Mode 0: Writing of 0 will be ignored. Always reads back a 0. 1: Force load event. This bit is intended to be used for testing and/or software force loading of the events in global load mode. Reset type: SYSRSn
1	OSHTCLR	R-0/W1S	0h	Clear One Shot latch 0: Writing of 0 will be ignored. Always reads back a 0. 1: Turns the one shot latch condition OFF. Reset type: SYSRSn
0	OSHTLD	R-0/W1S	0h	Enable Reload Event in One Shot Mode 0: Writing of 0 will be ignored. Always reads back a 0. 1: Turns the one shot latch condition ON. Upon occurrence of a chosen load strobe, one shadow to active reload occurs and the latch will be cleared. Hence writing 1 to this bit would allow one load strobe event to pass through and block further strobe events. Reset type: SYSRSn

15.12.3.39 GLDOSHTSTS Register (Offset = 7Ch) [Reset = 00000000h]

GLDOSHTSTS is shown in Figure 15-147 and described in Table 15-124.

Return to the Summary Table.

Global load one shot status register

Figure 15-147. GLDOSHTSTS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							OSHTLATCH
R-0-0h							R-0h

Table 15-124. GLDOSHTSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	OSHTLATCH	R	0h	One shot latch output 0: one shot latch condition is OFF. 1: one shot latch condition is ON. Reset type: SYSRSn

15.12.3.40 PWM1_CMPA Register (Offset = 80h) [Reset = 00000000h]

PWM1_CMPA is shown in Figure 15-148 and described in Table 15-125.

Return to the Summary Table.

PWM1 counter compare A register

Figure 15-148. PWM1_CMPA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPA															
R-0-0h																R/W-0h															

Table 15-125. PWM1_CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPA	R/W	0h	PWM1 Compare A register The value in the PWM1_CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM1 counter-compare module generates a 'time-base counter equal to CMP A' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM1A or the PWM1B output depending on the configuration of the PWM1_AQCTLA and PWM1_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM1_LOADAMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.3.41 PWM1_CMPAS Register (Offset = 82h) [Reset = 00000000h]

PWM1_CMPAS is shown in Figure 15-149 and described in Table 15-126.

Return to the Summary Table.

PWM1 counter compare A shadow register

Figure 15-149. PWM1_CMPAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPAS															
R-0-0h																R/W-0h															

Table 15-126. PWM1_CMPAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPAS	R/W	0h	PWM1 Compare A Shadow Register The value in the PWM1_CMPAS register is loaded into PWM!_CMPA register when shadow to active load occurs. Reset type: SYSRSn

15.12.3.42 PWM1_CMPB Register (Offset = 84h) [Reset = 00000000h]

PWM1_CMPB is shown in Figure 15-150 and described in Table 15-127.

Return to the Summary Table.

PWM1 counter compare B register

Figure 15-150. PWM1_CMPB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPB															
R-0-0h																R/W-0h															

Table 15-127. PWM1_CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPB	R/W	0h	PWM1 Compare B register The value in the PWM1_CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM1 counter-compare module generates a 'time-base counter equal to CMP B' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM1A or the PWM1B output depending on the configuration of the PWM1_AQCTLA and PWM1_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM1_LOADBMODE] field. By default this register is shadowed. Reset type: SYSRSn

15.12.3.43 PWM1_CMPBS Register (Offset = 86h) [Reset = 00000000h]

PWM1_CMPBS is shown in Figure 15-151 and described in Table 15-128.

Return to the Summary Table.

PWM1 counter compare B shadow register

Figure 15-151. PWM1_CMPBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPBS															
R-0-0h																R/W-0h															

Table 15-128. PWM1_CMPBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPBS	R/W	0h	PWM1 Compare B Shadow Register The value in the PWM1 CMPBS register is loaded into PWM1 CMPB register when shadow to active load occurs. Reset type: SYSRSn

15.12.3.44 PWM1_AQCTLA Register (Offset = 90h) [Reset = 00000000h]

PWM1_AQCTLA is shown in Figure 15-152 and described in Table 15-129.

Return to the Summary Table.

PWM1 action qualifier A register

Figure 15-152. PWM1_AQCTLA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-129. PWM1_AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-129. PWM1_AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.3.45 PWM1_AQCTLAS Register (Offset = 92h) [Reset = 00000000h]

PWM1_AQCTLAS is shown in Figure 15-153 and described in Table 15-130.

Return to the Summary Table.

PWM1 action qualifier A shadow register

Figure 15-153. PWM1_AQCTLAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-130. PWM1_AQCTLAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-130. PWM1_AQCTLAS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.3.46 PWM1_AQCTLB Register (Offset = 94h) [Reset = 00000000h]

PWM1_AQCTLB is shown in Figure 15-154 and described in Table 15-131.

Return to the Summary Table.

PWM1 action qualifier B register

Figure 15-154. PWM1_AQCTLB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-131. PWM1_AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-131. PWM1_AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.3.47 PWM1_AQCTLBS Register (Offset = 96h) [Reset = 00000000h]

PWM1_AQCTLBS is shown in Figure 15-155 and described in Table 15-132.

Return to the Summary Table.

PWM1 action qualifier B shadow register

Figure 15-155. PWM1_AQCTLBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	
R-0-0h				R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-132. PWM1_AQCTLBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 15-132. PWM1_AQCTLBS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

15.12.3.48 PWM1_AQSFRC Register (Offset = 98h) [Reset = 00000000h]

PWM1_AQSFRC is shown in Figure 15-156 and described in Table 15-133.

Return to the Summary Table.

PWM1 action qualifier software force

Figure 15-156. PWM1_AQSFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	PWMB			RESERVED	PWMA		
R-0-0h	R/W-0h			R-0-0h	R/W-0h		

Table 15-133. PWM1_AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6-4	PWMB	R/W	0h	Action qualifier software force on PWMB 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output B 010: Forces a continuous high on output B 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM1_AQOTSFRC[PWMB] = '1'. 110: Set (high) when PWM1_AQOTSFRC[PWMB] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM1_AQOTSFRC[PWMB] = '1'. Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved
2-0	PWMA	R/W	0h	Action qualifier software force on PWMA 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output A 010: Forces a continuous high on output A 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM1_AQOTSFRC[PWMA] = '1'. 110: Set (high) when PWM1_AQOTSFRC[PWMA] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM1_AQOTSFRC[PWMA] = '1'. Reset type: SYSRSn

15.12.3.49 PWM1_AQOTSFRC Register (Offset = 9Ah) [Reset = 00000000h]

PWM1_AQOTSFRC is shown in Figure 15-157 and described in Table 15-134.

Return to the Summary Table.

PWM1 action qualifier one time software force

Figure 15-157. PWM1_AQOTSFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			PWMB	RESERVED			PWMA
R-0-0h			R-0/W1S-0h	R-0-0h			R-0/W1S-0h

Table 15-134. PWM1_AQOTSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4	PWMB	R-0/W1S	0h	Action qualifier one time software force on PWMB 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMB. 1: Initiates a single software forced event Reset type: SYSRSn
3-1	RESERVED	R-0	0h	Reserved
0	PWMA	R-0/W1S	0h	Action qualifier one time software force on PWMA 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMA. 1: Initiates a single software forced event Reset type: SYSRSn

Chapter 16

Enhanced Capture (eCAP)

This chapter describes the enhanced capture (eCAP) module, which is used in systems where accurate timing of external events is important.

The enhanced capture (eCAP) module is a eCAP. See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with an eCAP module of the same type, to determine the differences between the types, and for a list of device-specific differences within a type.

16.1 Introduction

16.1.1 Features

The features of the eCAP module include:

- Speed measurements of rotating machinery (for example, toothed sprockets sensed by way of Hall sensors)
- Elapsed time measurements between position sensor pulses
- Period and duty cycle measurements of pulse train signals
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors

The eCAP module features described in this chapter include:

- 4-event time-stamp registers (each 32 bits)
- Edge polarity selection for up to four sequenced time-stamp capture events
- Interrupt on either of the four events
- Single-shot capture of up to four event time-stamps
- Continuous mode capture of time stamps in a four-deep circular buffer
- Absolute time-stamp capture
- Difference (Delta) mode time-stamp capture
- When not used in capture mode, the eCAP module can be configured as a single-channel PWM output

The capture functionality of the Type 1 eCAP is enhanced from the Type 0 eCAP with the following added features:

- Event filter reset bit:
  - Writing a 1 to ECCTL2[CTRFILTRESET] clears the event filter, the modulo counter, and any pending interrupts flags. Resetting the bit is useful for initialization and debug.
- Modulo counter status bits:
  - The modulo counter (ECCTL2 [MODCNTRSTS]) indicates which capture register is loaded next. In the Type 0 eCAP, to know the current state of the modulo counter was not possible
- Input multiplexer:
  - ECCTL0 [INPUTSEL] selects one of 128 input signals, which are detailed in Section 16.3.
- EALLOW protection:
  - EALLOW protection was added to critical registers. To maintain software compatibility with Type-0, configure DEV_CFG_REGS.ECAPTYPE to make these registers unprotected.

The capture functionality of the Type 2 eCAP is enhanced from the Type 1 eCAP with the following added features:

- Added ECAPxSYNCINSEL register:
  - ECAPxSYNCINSEL register is added for each eCAP to select an external SYNCIN. Every eCAP can have a separate SYNCIN signal.

16.1.2 ECAP Related Collateral

Foundational Materials

- C28x Academy - ECAP

16.2 Description

The eCAP module represents one complete capture channel that can be instantiated multiple times, depending on the target device. In the context of this guide, one eCAP channel has the following independent key resources:

- Capture inputs can be connected using the Input X-BAR
- 128:1 input multiplexer
- Output X-BAR is used to configure output in APWM mode
- 32-bit time base (counter)

- 4 × 32-bit time-stamp capture registers (CAP1-CAP4)
- Four-stage sequencer (modulo4 counter) that is synchronized to external events, eCAP pin rising/falling edges.
- Modulo counter status register (MODCNTRSTS) to indicate sequencer state
- Independent edge polarity (rising/falling edge) selection for all four events
- Input capture signal prescaling (from 2-62 or bypass)
- One-shot compare register (two bits) to freeze captures after 1-4 time-stamp events
- Control for continuous time-stamp captures using a four-deep circular buffer (CAP1-CAP4) scheme
- Ability to reset event filter, modulo counter, and interrupt flags
- Interrupt capabilities on any of the four capture events
- EALLOW protection to control registers

16.3 Configuring Device Pins for the eCAP

The Input X-BAR connects the device pins to the module as input. Any GPIO on the device can be configured as an input. The GPIO input qualification can be set to synchronous or asynchronous mode by setting the GPxQSELn register bits. Using synchronized inputs can help with noise immunity but affects the eCAP accuracy by ±2 cycles. The internal pull-ups can be configured in the GPyPUD register. Since the GPIO mode is used, the GPyINV register can invert the signals.

A 128:1 input multiplexer must also be configured (see Figure 16-3). This multiplexer can select a variety of inputs by configuring ECCTL0.INPUTSEL.

Table 16-1. eCAP Input Selection

Selection of ECAP Input	ECAP1 INDEX	ECAP2 INDEX
INPUTXBAR1	0	0
INPUTXBAR2	1	1
INPUTXBAR3	2	2
INPUTXBAR4	3	3
INPUTXBAR5	4	4
INPUTXBAR6	5	5
INPUTXBAR7	6	6
INPUTXBAR8	7	7
INPUTXBAR9	8	8
INPUTXBAR10	9	9
INPUTXBAR11	10	10
INPUTXBAR12	11	11
INPUTXBAR13	12	12
INPUTXBAR14	13	13
INPUTXBAR15	14	14
INPUTXBAR16	15	15
	16	16
OUTPUTXBAR1	17	17
OUTPUTXBAR2	18	18
OUTPUTXBAR3	19	19
OUTPUTXBAR4	20	20
OUTPUTXBAR5	21	21

Table 16-1. eCAP Input Selection (continued)

Selection of ECAP Input	ECAP1 INDEX	ECAP2 INDEX
OUTPUTXBAR6	22	22
OUTPUTXBAR7	23	23
OUTPUTXBAR8	24	24
ADCAEVT1	25	25
ADCAEVT2	26	26
ADCAEVT3	27	27
	28	28
	29	29
	30	30
CMPSS1_CTRIPL	31	31
CMPSS2_CTRIPL	32	32
CMPSS3_CTRIPL	33	33
	34	34
CMPSS1_CTRIPH	35	35
CMPSS2_CTRIPH	36	36
CMPSS3_CTRIPH	37	37
	38	38
	39	39
	40	40
	41	41
	42	42
CMPSS1_CTRIPH_OR_CTRIPL	43	43
CMPSS2_CTRIPH_OR_CTRIPL	44	44
CMPSS3_CTRIPH_OR_CTRIPL	45	45
	46	46
INPUTXBAR7	47	Reserved
INPUTXBAR8	Reserved	47
Reserved	48-127	48-127

The Output X-BAR must be used to connect output signals to the OUTPUTXBARx output locations. The GPIO mux must then be configured to connect the OUTPUTXBARx lines to any of several IO pins with the GPIO mux. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux, GPIO settings, and X-BAR configuration.

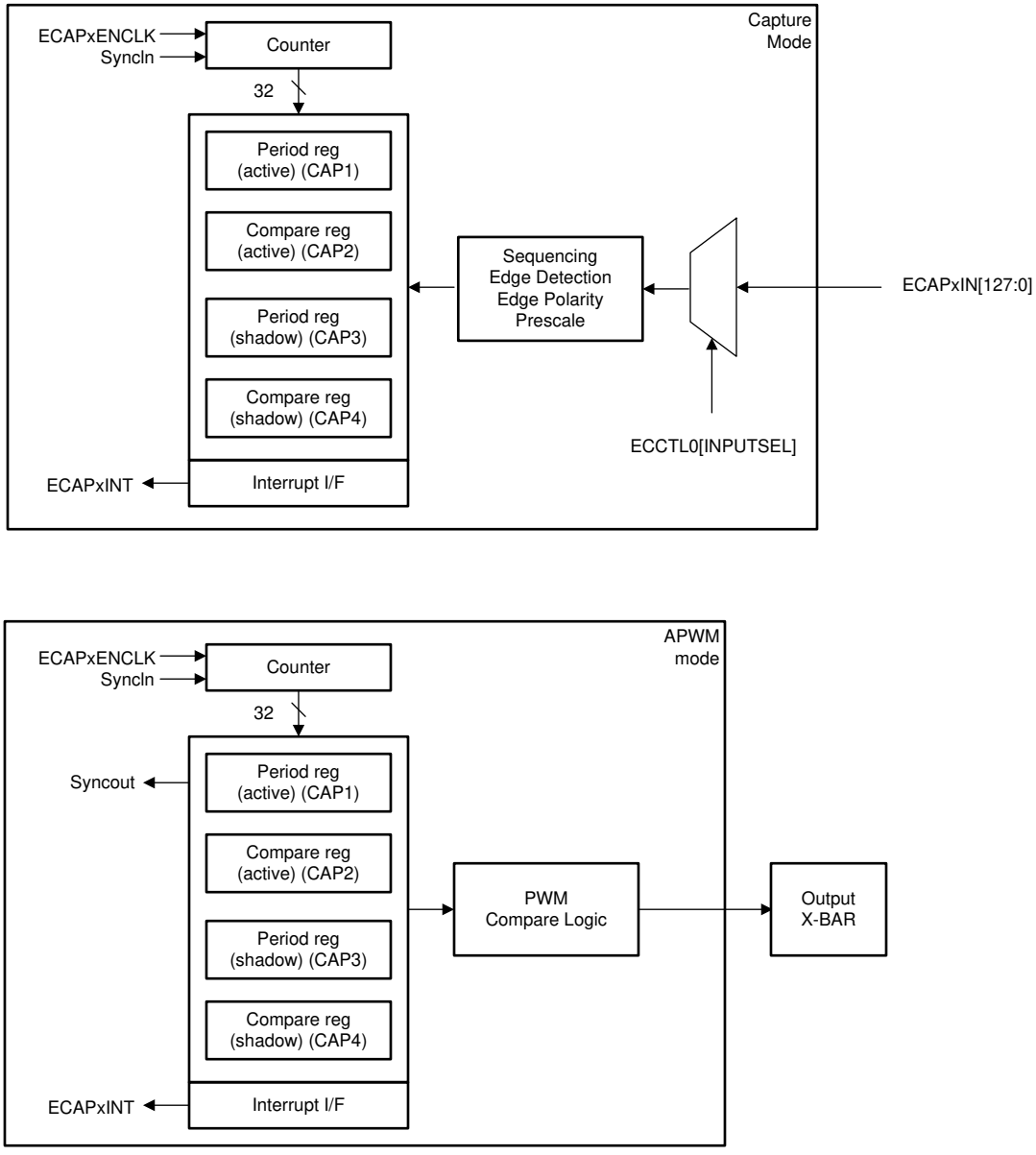
Note

ECAPxIN has to be at least 2 * SYSCLK-cycles wide to be properly captured by the eCAP module; otherwise, the input pulse can get missed from sampling by the SYSCLK .

16.4 Capture and APWM Operating Mode

Use the eCAP module resources to implement a single-channel PWM generator (with 32-bit capabilities) when the eCAP module is not being used for input captures. The counter operates in count-up mode, providing a time-base for asymmetrical pulse width modulation (PWM) waveforms. The CAP1 and CAP2 registers become the active period and compare registers, respectively, while CAP3 and CAP4 registers become the period and compare shadow registers, respectively. Figure 16-1 is a high-level view of both the capture and auxiliary pulse-width modulator (APWM) modes of operation.

Figure 16-2 further descries the output of the eCAP in APWM mode based on the CMP and PRD values.



A. A single pin is shared between CAP and APWM functions. In capture mode, the pin is an input; in APWM mode, the pin is an output.

B. In APWM mode, writing any value to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.

Figure 16-1. Capture and APWM Modes of Operation

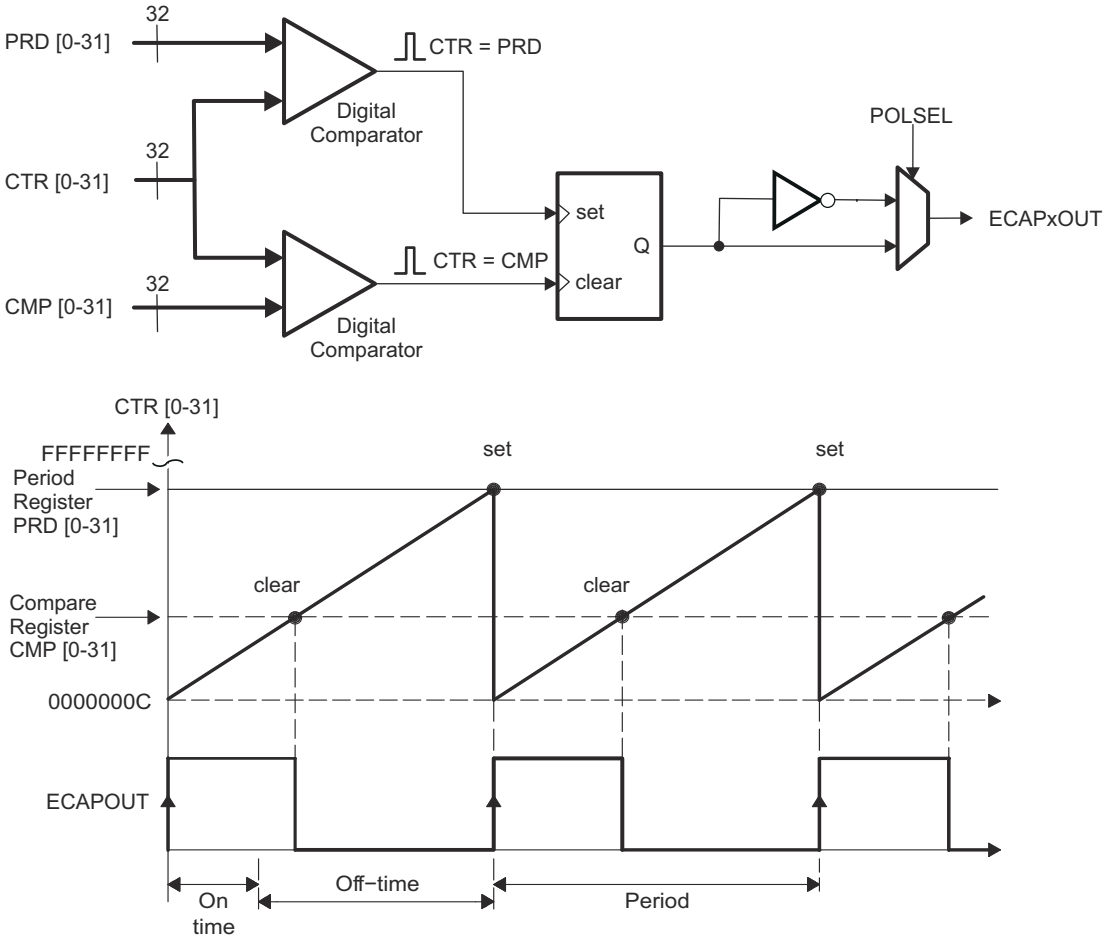


Figure 16-2. Counter Compare and PRD Effects on the eCAP Output in APWM Mode

16.5 Capture Mode Description

Figure 16-3 shows the various components that implement the capture function.

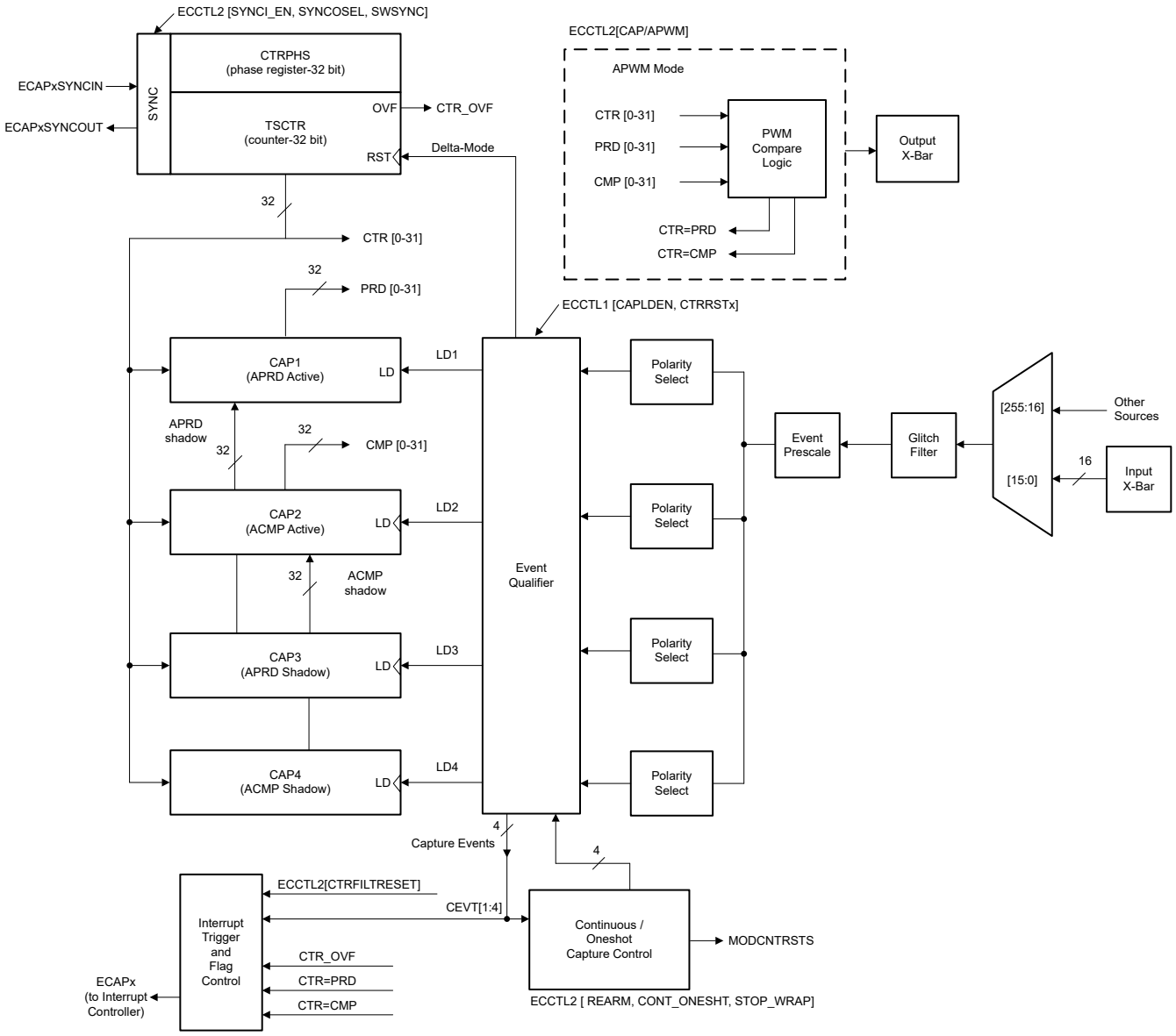
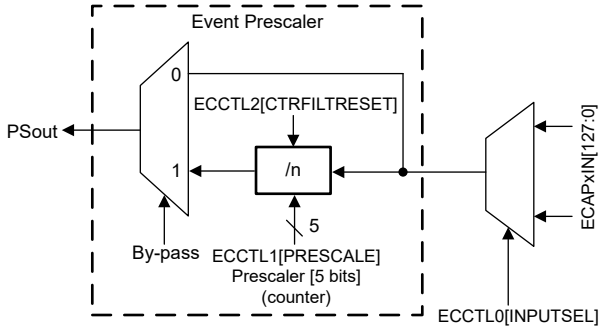


Figure 16-3. eCAP Block Diagram

16.5.1 Event Prescaler

An input capture signal (pulse train) can be prescaled by N = 2-62 (in multiples of 2) or can bypass the prescaler. This is useful when very high frequency signals are used as inputs. Figure 16-4 shows a functional diagram and Figure 16-5 shows the operation of the prescale function. The event prescaler can be reset by setting the ECCTL2.CTRFILTRESET register bit.



A. When a prescale value of 1 is chosen (ECCTL1[13:9] = 0,0,0,0,0), the input capture signal bypasses the prescale logic completely.
B. The first Rise edge after Prescale configuration change is not passed to Capture logic, prescaler value takes into effect on the second rising edge after the configuration.

Figure 16-4. Event Prescale Control

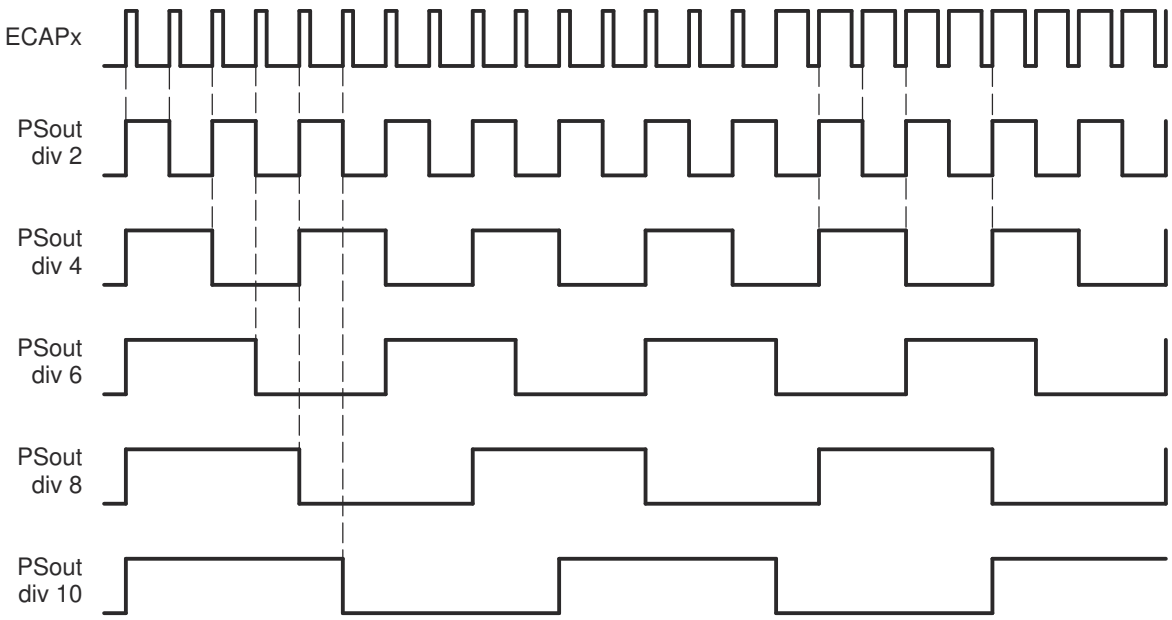


Figure 16-5. Prescale Function Waveforms

16.5.2 Edge Polarity Select and Qualifier

Functionality and features include:

- Four independent edge polarity (rising edge/falling edge) selection muxes are used, one for each capture event.
- Each edge (up to 4) is event qualified by the Modulo4 sequencer.
- The edge event is gated to the respective CAPx register by the Mod4 counter. The CAPx register is loaded on the falling edge.

16.5.3 Continuous/One-Shot Control

Operation of eCAP in Continuous/One-Shot mode:

- The Mod4 (2-bit) counter is incremented using edge qualified events (CEVT1-CEVT4).
- The Mod4 counter continues counting (0->1->2->3->0) and wraps around unless stopped.
- During one-shot operation, a 2-bit stop register (STOP_WRAP) is used to compare the Mod4 counter output, and when equal, stops the Mod4 counter and inhibits further loads of the CAP1-CAP4 registers. In this mode, if TSCCTR counter is configured to reset on capture event (CEVTx) by configuring ECCTL1.CTRRSTx bit, the operation still keeps resetting the TSCCTR counter on capture event (CEVTx) after the STOP_WRAP value is reached and re-arm (REARM) has not occurred.

The continuous/one-shot block controls the start, stop and reset (zero) functions of the Mod4 counter, using a mono-shot type of action that can be triggered by the stop-value comparator and re-armed using software control.

Once armed, the eCAP module waits for 1-4 (defined by stop-value) capture events before freezing both the Mod4 counter and contents of CAP1-4 registers (time stamps).

Re-arming prepares the eCAP module for another capture sequence. Also, re-arming clears (to zero) the Mod4 counter and permits loading of CAP1-4 registers again, providing the CAPLDEN bit is set.

In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to CAP1-4 in a circular buffer sequence.

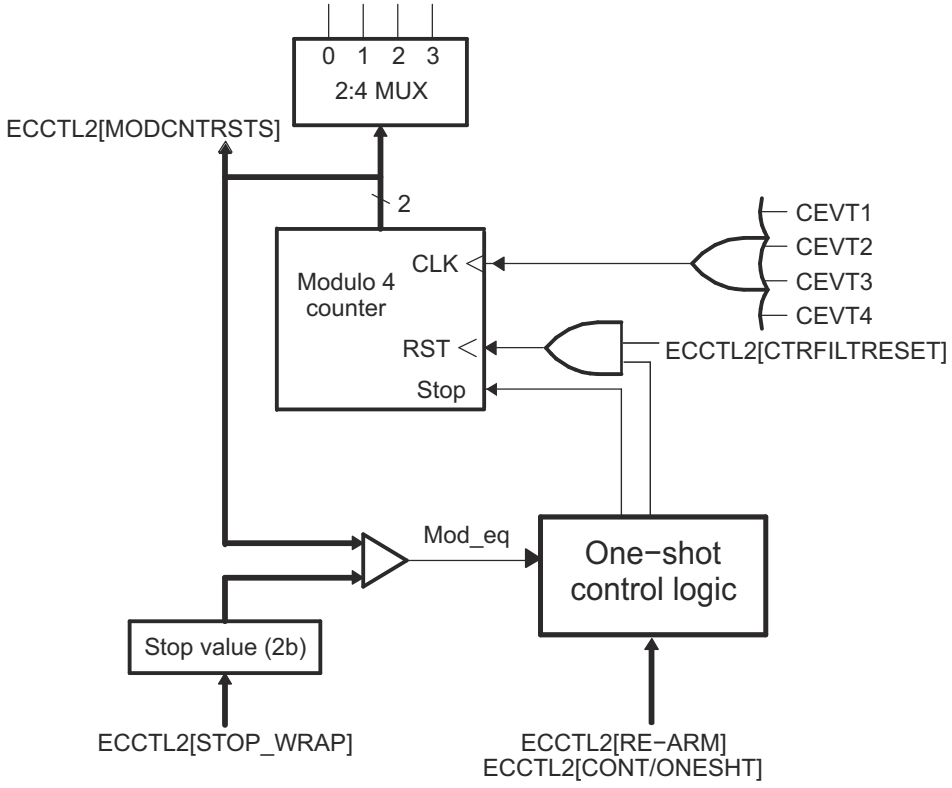


Figure 16-6. Details of the Continuous/One-shot Block

16.5.4 32-Bit Counter and Phase Control

This counter provides the time-base for event captures, and is clocked using the system clock.

A phase register is provided to achieve synchronization with other counters using a hardware and software forced sync. This is useful in APWM mode when a phase offset between modules is needed.

On any of the four event loads, an option to reset the 32-bit counter is given. This is useful for time difference capture. The 32-bit counter value is captured first, then the counter value is reset to 0 by any of the LD1-LD4 signals.

16.5.5 CAP1-CAP4 Registers

These 32-bit registers are supplied by the 32-bit counter timer bus, CTR[0-31], and are loaded (capture a time-stamp) when the respective LD inputs are strobed.

Control bit CAPLDEN can inhibit loading of the capture registers. During one-shot operation, this bit is cleared (loading is inhibited) automatically when a stop condition occurs, StopValue = Mod4.

CAP1 and CAP2 registers become the active period and compare registers, respectively, in APWM mode.

CAP3 and CAP4 registers become the respective shadow registers (APRD and ACMP) for CAP1 and CAP2 during APWM operation.

16.5.6 eCAP Synchronization

eCAP modules can be synchronized with each other by selecting a common SYNCIN source. SYNCIN source for eCAP can be either software sync-in or external sync-in. The external sync-in signal can come from eCAP, X-Bar, or MCPWM. The SWSYNC of the eCAP module is logical ORed with the SYNC signal as shown in Figure 16-7. The SYNC signal is defined by the selection of ECAPxSYNCINSEL[SEL] as shown in Figure 16-8.

Note

ECAPxSYNCOUT going to the ECAPSYNCIN multiplexer is disabled. For example, ECAP1SYNCOUT cannot be a sync in to ECAP1, but ECAP1SYNCOUT can be a sync in to ECAP2, ECAP3, and so on.

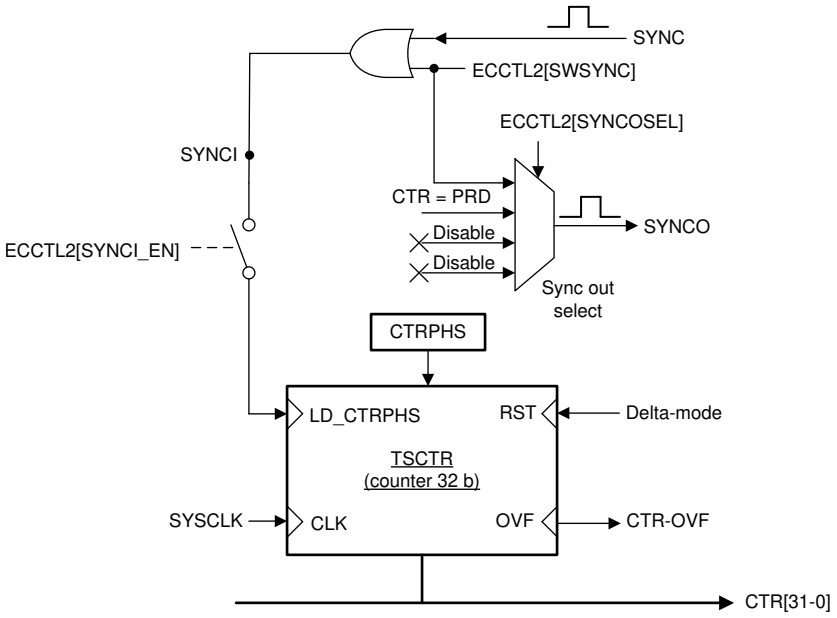


Figure 16-7. Details of the Counter and Synchronization Block

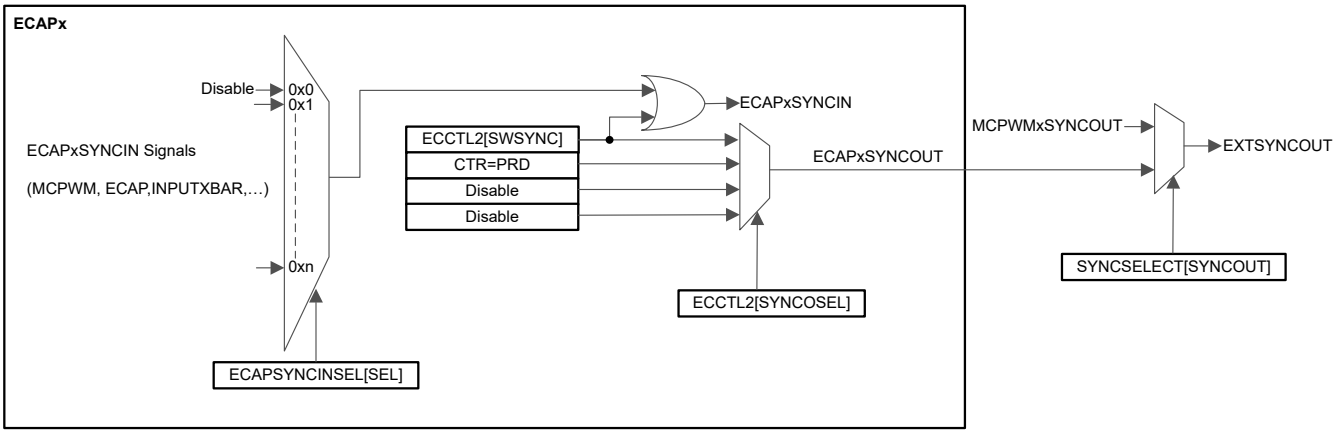


Figure 16-8. eCAP Synchronization Scheme

16.5.6.1 Example 1 - Using SWSYNC with ECAP Module

Implement the following steps to use SWSYNC with ECAP1 and ECAP2.

- Configure ECAP[1..2].ECAPSYNCINSEL.SEL = 0x0 to disable external SYNCIN coming to eCAP1.
- Configure ECAP[1..2].ECCTL2.SWSYNC = 0x1, to force Software Synchronization of the TSCTR counter.

To use SWSYNC with other eCAP modules, make sure that the previous eCAP chain is not generating a SYNCOUT signal that interferes with the software synchronization.

16.5.7 Interrupt Control

Operation and features of the eCAP interrupt control include (see Figure 16-9):

- An interrupt can be generated on capture events (CEVT1-CEVT4, CTROVF) or APWM events (CTR = PRD, CTR = CMP).
- A counter overflow event (FFFFFFFF->00000000) is also provided as an interrupt source (CTROVF).
- The capture events are edge and sequencer-qualified (ordered in time) by the polarity select and Mod4 gating, respectively.
- One of these events can be selected as the interrupt source (from the eCAPx module) going to the PIE.
- Seven interrupt events (CEVT1, CEVT2, CEVT3, CEVT4, CNTOVF, CTR=PRD, CTR=CMP) can be generated.
- The interrupt enable register (ECEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (ECFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated to the PIE only if any of the interrupt events are enabled, the flag bit is 1, and the INT flag bit is 0. The interrupt service routine must clear the global interrupt flag bit and the serviced event using the interrupt clear register (ECCLR) before any other interrupt pulses are generated. All interrupt flags are cleared upon an event filter reset by writing a 1 to ECCTL2[CLRFILTRESET]. To force an interrupt event, use the interrupt force register (ECFRC). This is useful for test purposes.

Note

The CEVT1, CEVT2, CEVT3, CEVT4 flags are only active in capture mode (ECCTL2[CAP/APWM == 0]). The CTR=PRD, CTR=CMP flags are only valid in APWM mode (ECCTL2[CAP/APWM == 1]). CNTOVF flag is valid in both modes.

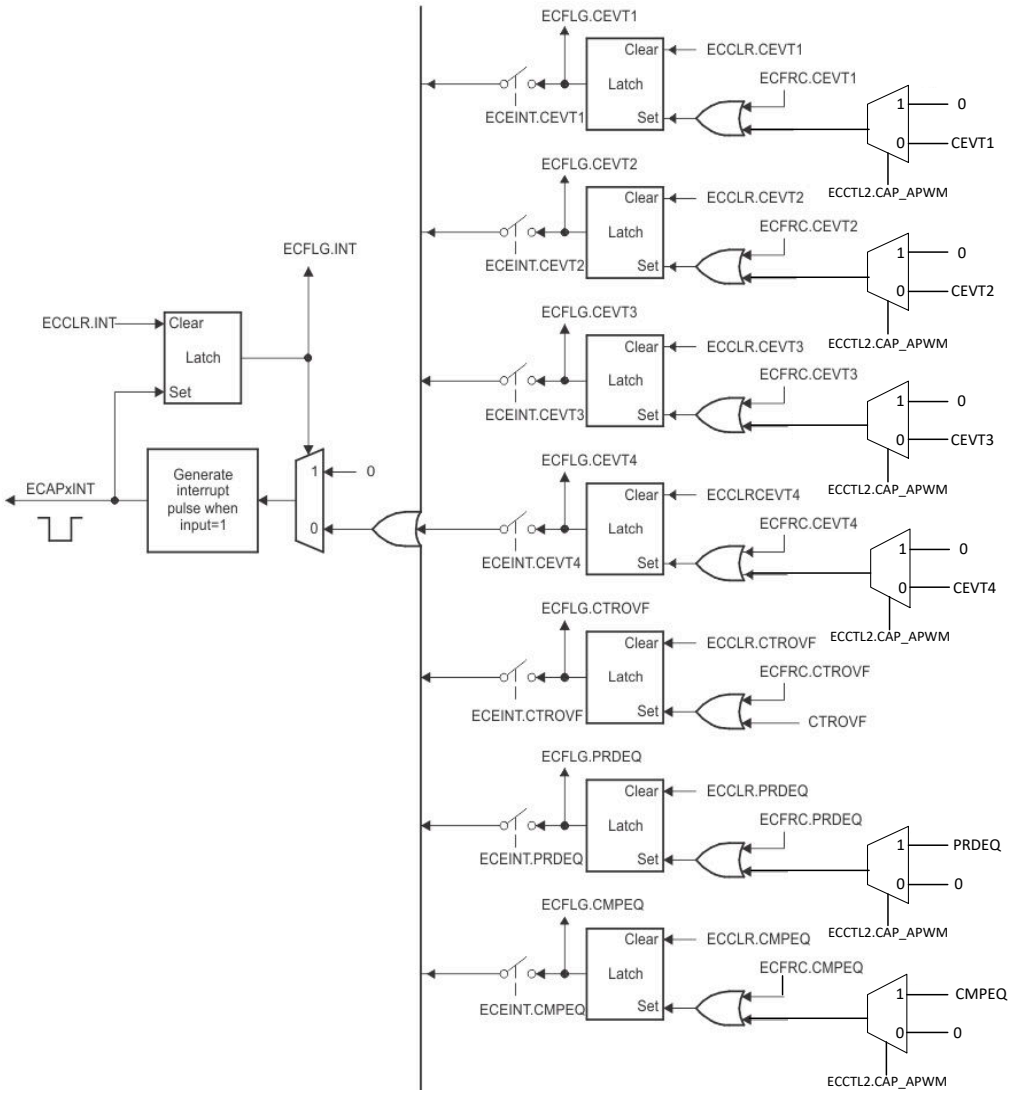


Figure 16-9. Interrupts in eCAP Module

16.5.8 Shadow Load and Lockout Control

In capture mode, this logic inhibits (locks out) any shadow loading of CAP1 or CAP2 from APRD and ACMP registers, respectively.

In APWM mode, shadow loading is active and two choices are permitted:

- Immediate - APRD or ACMP are transferred to CAP1 or CAP2 immediately upon writing a new value.
- On period equal, CTR[31:0] = PRD[31:0].

16.5.9 APWM Mode Operation

Main operating highlights of the APWM section:

- The time-stamp counter bus is made available for comparison by way of 2 digital (32-bit) comparators.
- When CAP1/2 registers are not used in capture mode, the contents can be used as Period and Compare values in APWM mode.
- Double buffering is achieved using shadow registers APRD and ACMP (CAP3/4). The shadow register contents are transferred over to CAP1/2 registers, either immediately upon a write, or on a CTR = PRD trigger.
- In APWM mode, writing to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.
- During initialization, write to the active registers for both period and compare. This automatically copies the initial values into the shadow values. For subsequent compare updates during run-time, use the shadow registers.

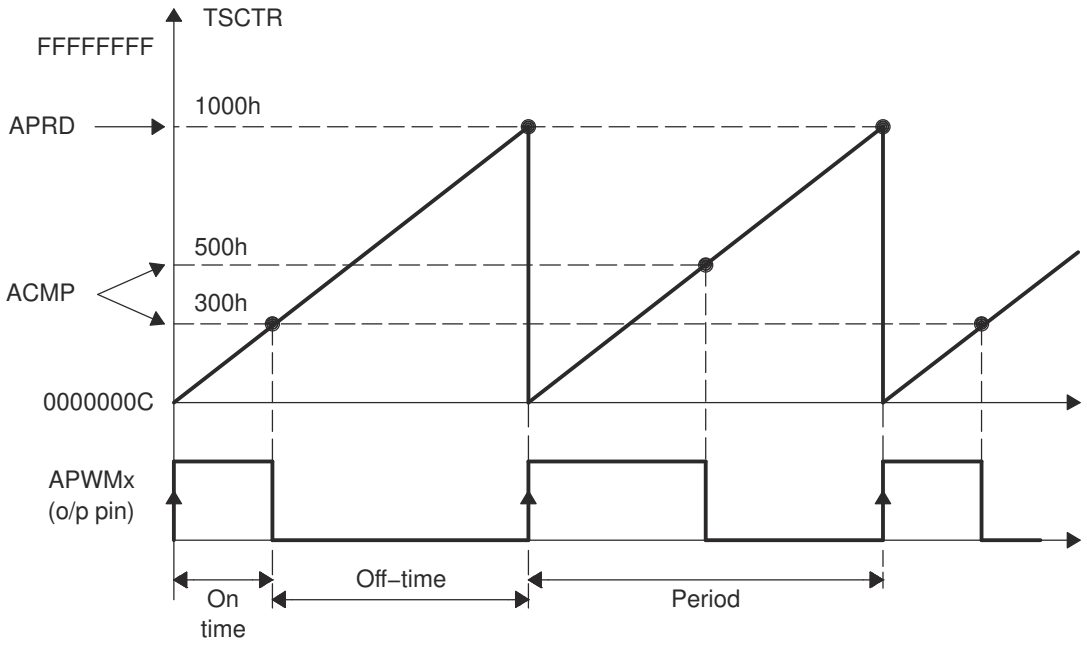


Figure 16-10. PWM Waveform Details Of APWM Mode Operation

The behavior of APWM active high mode (APWMPOL == 0) is as follows:

CMP = 0x00000000, output low for duration of period (0% duty)

CMP = 0x00000001, output high 1 cycle

CMP = 0x00000002, output high 2 cycles

CMP = PERIOD, output high except for 1 cycle (<100% duty)

CMP = PERIOD+1, output high for complete period (100% duty)

CMP > PERIOD+1, output high for complete period

The behavior of APWM active low mode (APWMPOL == 1) is as follows:

CMP = 0x00000000, output high for duration of period (0% duty)

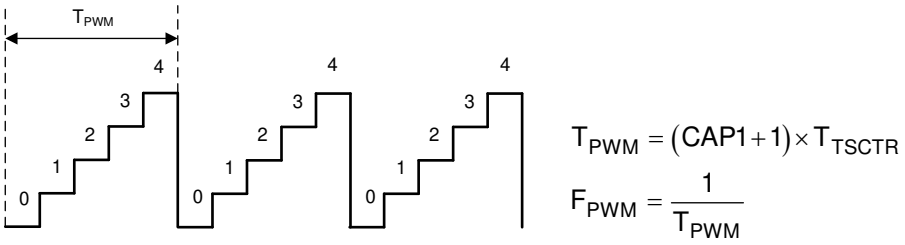
CMP = 0x00000001, output low 1 cycle

CMP = 0x00000002, output low 2 cycles

CMP = PERIOD, output low except for 1 cycle (<100% duty)

CMP = PERIOD+1, output low for complete period (100% duty)

CMP > PERIOD+1, output low for complete period



$$T_{PWM} = (CAP1 + 1) \times T_{TSCTR}$$

$$F_{PWM} = \frac{1}{T_{PWM}}$$

Figure 16-11. Time-Base Frequency and Period Calculation

16.6 Application of the eCAP Module

The following sections provide applications examples to show how to operate the eCAP module.

16.6.1 Example 1 - Absolute Time-Stamp Operation Rising-Edge Trigger

Figure 16-12 shows an example of continuous capture operation (Mod4 counter wraps around). In this figure, TSCTR counts-up without resetting and capture events are qualified on the rising edge only, this gives period (and frequency) information.

On an event, the TSCTR contents (time-stamp) is first captured, then Mod4 counter is incremented to the next state. When the TSCTR reaches FFFFFFFF (maximum value), the Mod4 counter wraps around to 00000000 (not shown in Figure 16-12), if this occurs, the CTROVF (counter overflow) flag is set, and an interrupt (if enabled) occurs. Captured Time-stamps are valid at the point indicated by the diagram (after the fourth event); hence, event CEVT4 can conveniently be used to trigger an interrupt and the CPU can read data from the CAPx registers.

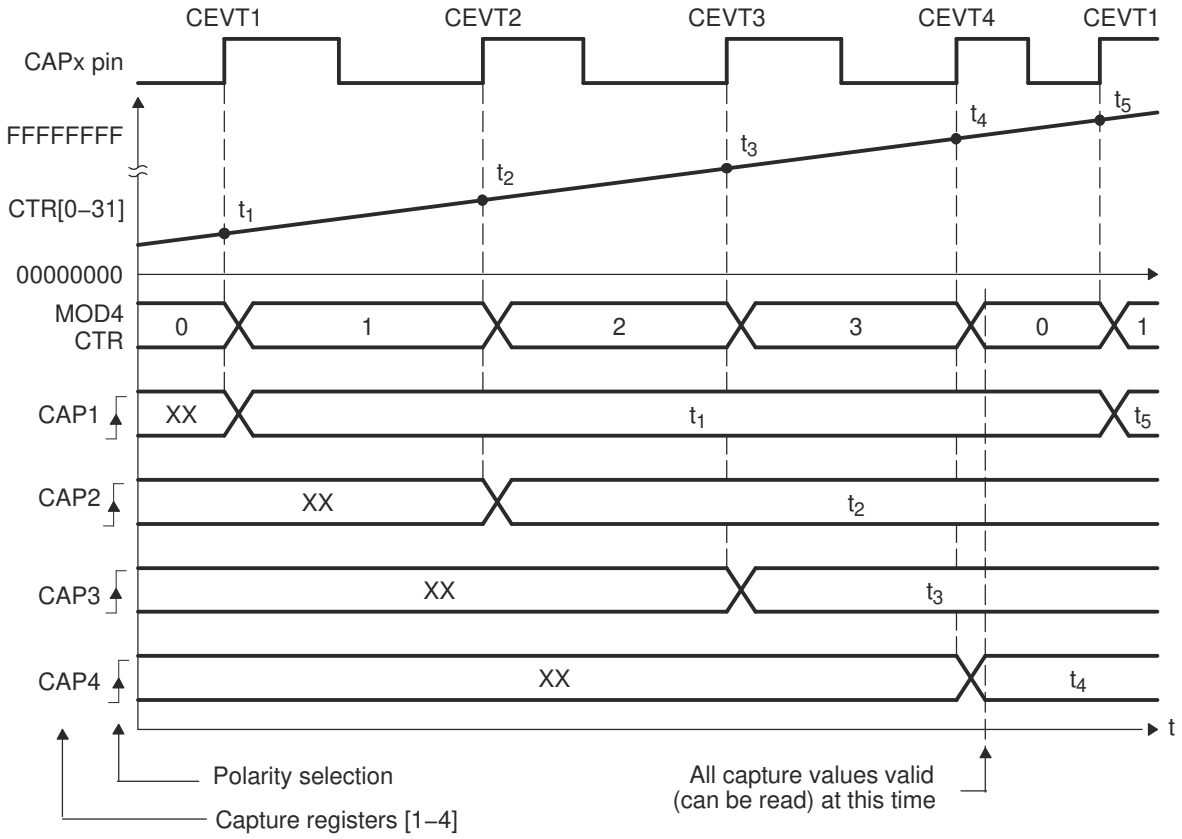


Figure 16-12. Capture Sequence for Absolute Time-stamp and Rising-Edge Detect

16.6.2 Example 2 - Absolute Time-Stamp Operation Rising- and Falling-Edge Trigger

In Figure 16-13, the eCAP operating mode is almost the same as in the previous section except capture events are qualified as either rising or falling edge, this now gives both period and duty cycle information, that is: Period1 = $t_3 - t_1$, Period2 = $t_5 - t_3$, …and so on. Duty Cycle1 (on-time %) = $(t_2 - t_1)$ / Period1 x 100%, and so on. Duty Cycle1 (off-time %) = $(t_3 - t_2)$ / Period1 x 100%, and so on.

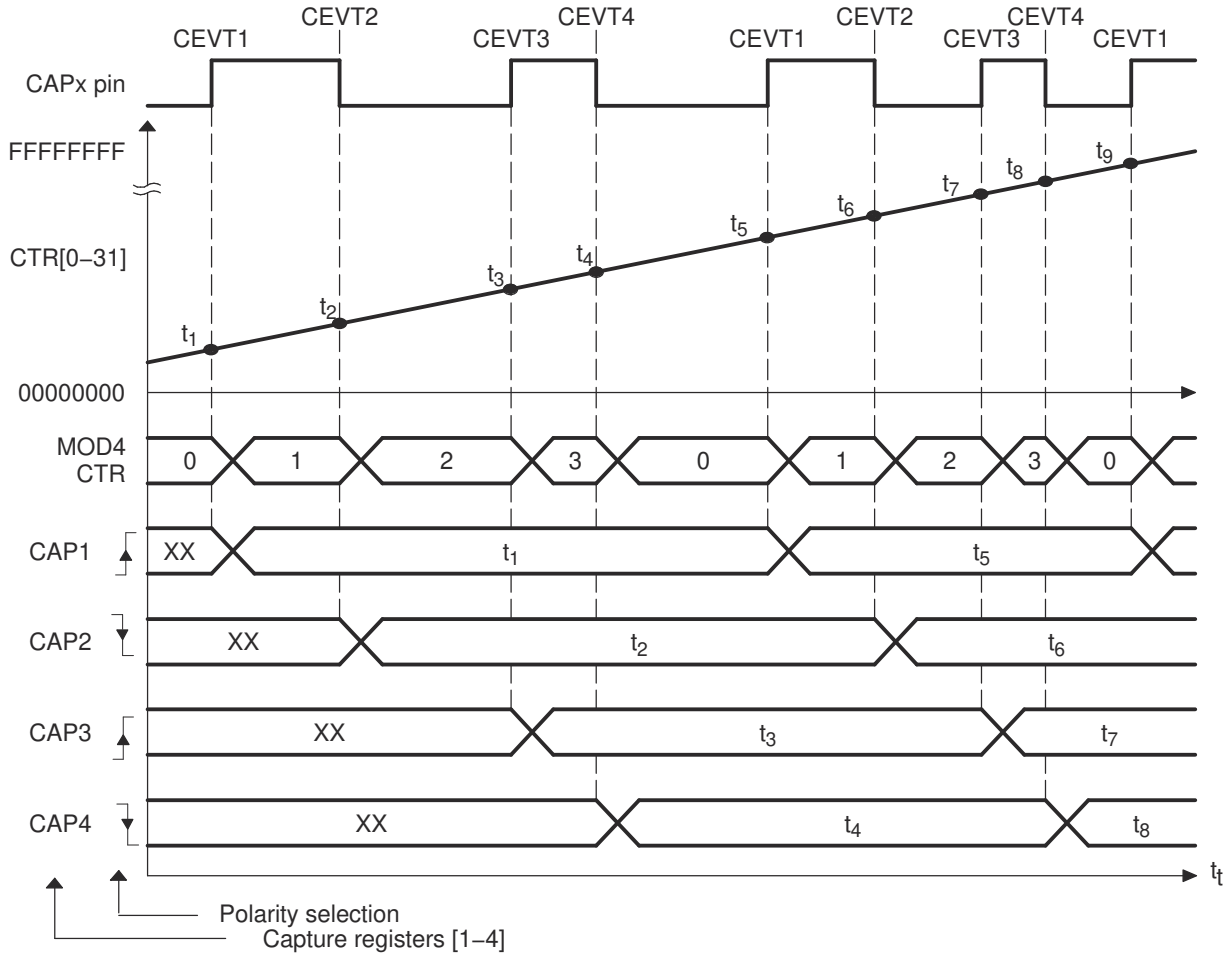


Figure 16-13. Capture Sequence for Absolute Time-stamp with Rising- and Falling-Edge Detect

16.6.3 Example 3 - Time Difference (Delta) Operation Rising-Edge Trigger

Figure 16-14 shows how the eCAP module can be used to collect delta timing data from pulse train waveforms. Here Continuous Capture mode (TSCTR counts-up without resetting, and Mod4 counter wraps around) is used. In Delta-time mode, TSCTR is reset back to zero on every valid event. Here capture events are qualified as rising edge only. On an event, TSCTR contents (Time-Stamp) is captured first, and then TSCTR is reset to zero. The Mod4 counter then increments to the next state. If TSCTR reaches FFFFFFFF (maximum value), before the next event, the Mod4 counter wraps around to 00000000 and continues, a CNTOVF (counter overflow) flag is set, and an interrupt (if enabled) occurs. The advantage of Delta-time mode is that the CAPx contents directly give timing data without the need for CPU calculations, that is, Period1 = T_1, Period2 = T_2, and so on. As shown in Figure 16-14, the CEVT1 event is a good trigger point to read the timing data, T_1, T_2, T_3, T_4 are all valid here.

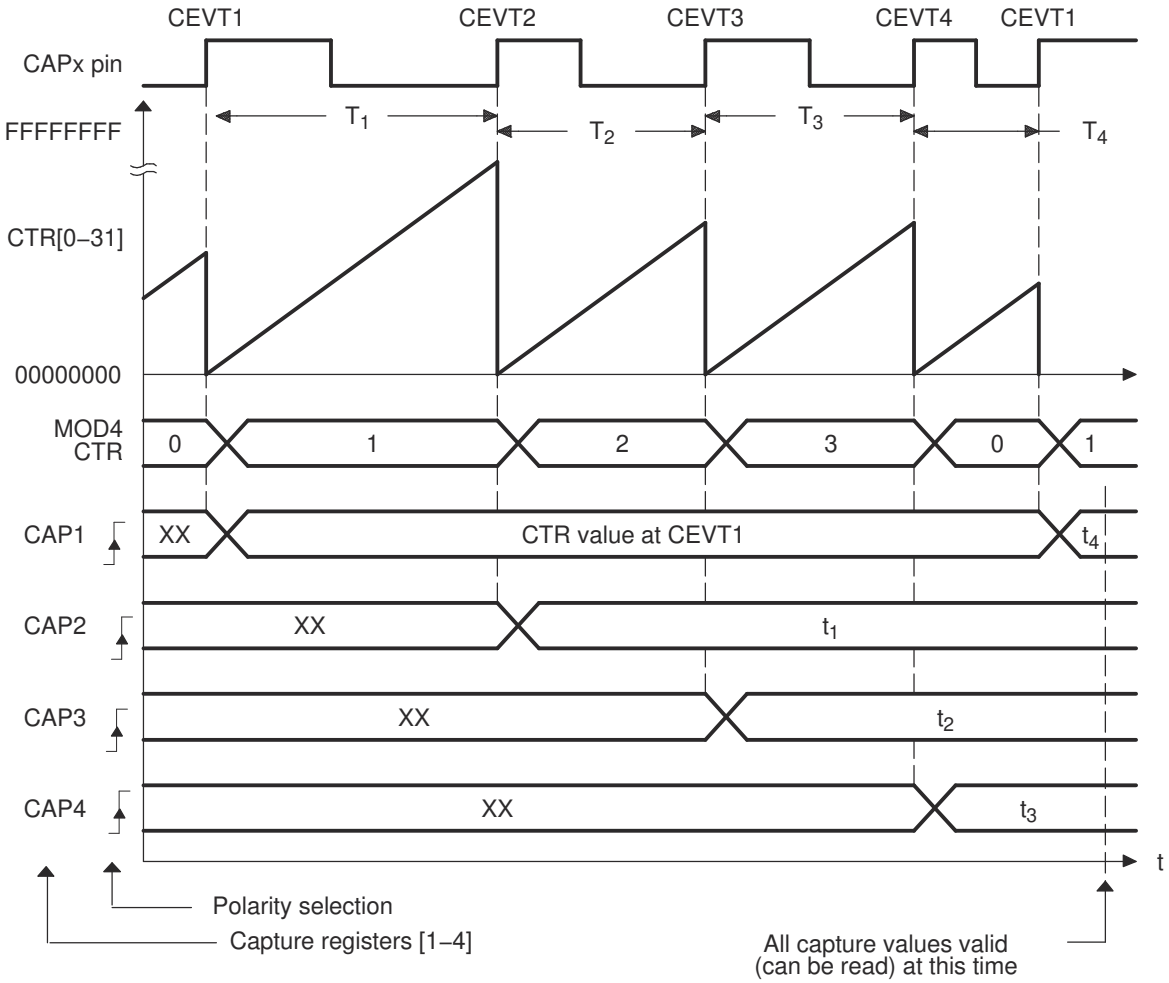


Figure 16-14. Capture Sequence for Delta Mode Time-stamp and Rising Edge Detect

16.6.4 Example 4 - Time Difference (Delta) Operation Rising- and Falling-Edge Trigger

In Figure 16-15, the eCAP operating mode is almost the same as in previous section except capture events are qualified as either rising or falling edge, this now gives both period and duty cycle information, that is: Period1 = T_1+T_2, Period2 = T_3+T_4, and so on. Duty Cycle1 (on-time %) = T_1 / Period1 x 100%, Duty Cycle1 (off-time %) = T_2 / Period1 x 100%, and so on.

During initialization, write to the active registers for both period and compare. This action automatically copies the init values into the shadow values. For subsequent compare updates during run-time, the shadow registers must be used.

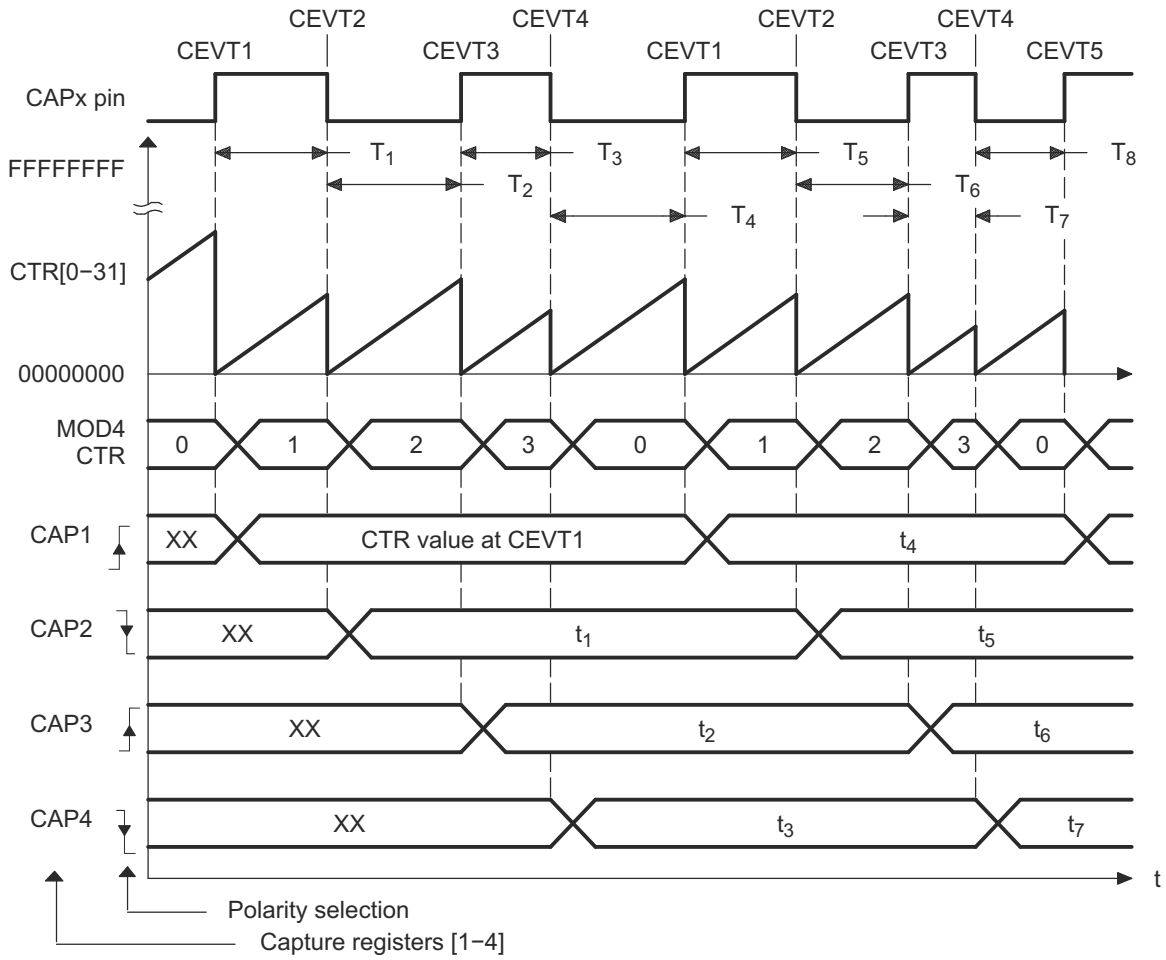


Figure 16-15. Capture Sequence for Delta Mode Time-stamp with Rising- and Falling-Edge Detect

16.7 Application of the APWM Mode

In this example, the eCAP module is configured to operate as a PWM generator. Here, a very simple single-channel PWM waveform is generated from the APWMx output pin. The PWM polarity is active high, which means that the compare value (CAP2 reg is now a compare register) represents the on-time (high level) of the period. Alternatively, if the APWMPOL bit is configured for active low, then the compare value represents the off-time.

16.7.1 Example 1 - Simple PWM Generation (Independent Channels)

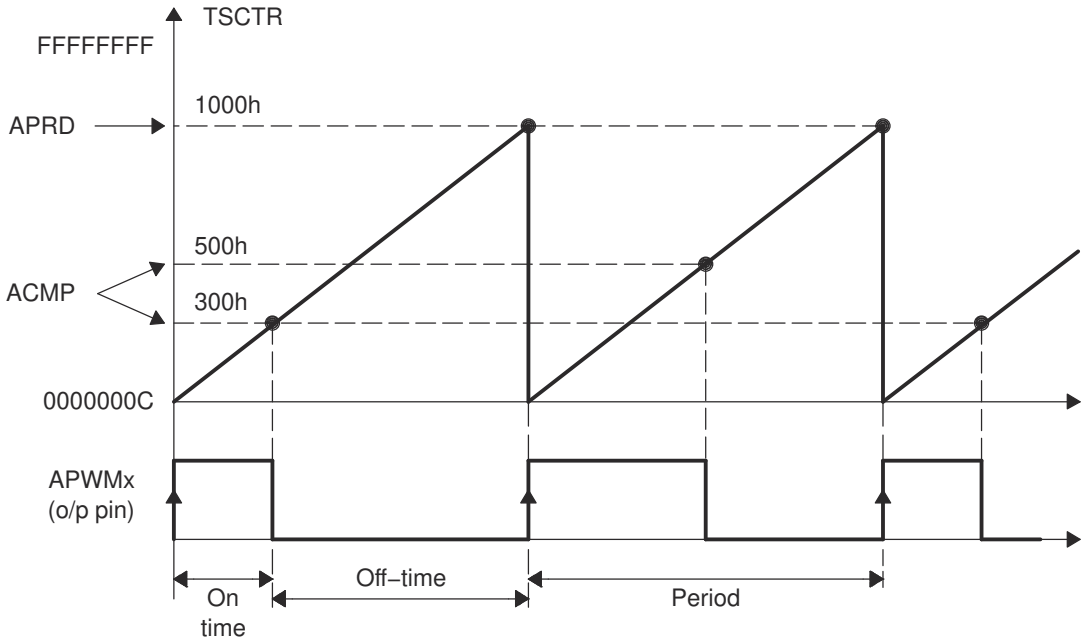


Figure 16-16. PWM Waveform Details of APWM Mode Operation

16.8 Software

16.8.1 ECAP Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/ecap

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

16.8.1.1 eCAP APWM Example

FILE: ecap_ex1_apwm.c

This program sets up the eCAP module in APWM mode. The PWM waveform will come out on GPIO5. The frequency of PWM is configured to vary between 5Hz and 10Hz using the shadow registers to load the next period/compare values.

16.8.1.2 eCAP Capture PWM Example

FILE: ecap_ex2_capture_pwm.c

This example configures MCPWM1A for:

- Up count mode
- Period starts at 500 and goes up to 8000
- Toggle output on PRD

eCAP1 is configured to capture the time between rising and falling edge of the MCPWM1A output.

External Connections

- eCAP1 is on GPIO2
- MCPWM1A is on GPIO0
- Connect GPIO0 to GPIO2.

Watch Variables

- *ecap1PassCount* - Successful captures.
- *ecap1IntCount* - Interrupt counts.

16.9 ECAP Registers

This Section describes the ECAP Registers.

16.9.1 ECAP Base Address Table

Table 16-2. ECAP Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
ECap1Regs	ECAP_REGS	ECAP1_BASE	0x0000_5200	YES	YES

16.9.2 ECAP_REGS Registers

Table 16-3 lists the memory-mapped registers for the ECAP_REGS registers. All register offset addresses not listed in Table 16-3 should be considered as reserved locations and the register contents should not be modified.

Table 16-3. ECAP_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	TSCTR	Time-Stamp Counter	
2h	CTRPHS	Counter Phase Offset Value Register	
4h	CAP1	Capture 1 Register	
6h	CAP2	Capture 2 Register	
8h	CAP3	Capture 3 Register	
Ah	CAP4	Capture 4 Register	
12h	ECCTL0	Capture Control Register 0	EALLOW
14h	ECCTL1	Capture Control Register 1	EALLOW
15h	ECCTL2	Capture Control Register 2	EALLOW
16h	ECEINT	Capture Interrupt Enable Register	EALLOW
17h	ECFLG	Capture Interrupt Flag Register	
18h	ECCLR	Capture Interrupt Clear Register	
19h	ECFRC	Capture Interrupt Force Register	EALLOW
1Eh	ECAPSYNCINSEL	SYNC source select register	EALLOW

Complex bit access types are encoded to fit into small table cells. Table 16-4 shows the codes that are used for access types in this section.

Table 16-4. ECAP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

16.9.2.1 TSCTR Register (Offset = 0h) [Reset = 00000000h]

TSCTR is shown in Figure 16-17 and described in Table 16-5.

Return to the Summary Table.

Time-Stamp Counter

Figure 16-17. TSCTR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSCTR																															
R/W-0h																															

Table 16-5. TSCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TSCTR	R/W	0h	Active 32-bit counter register that is used as the capture time-base HR mode : 1) This register reads HRCOUNTER value and is not writable 2) can be reset using CTRFILTRESET 3) Its not synchronized to SYSCLK domain so reads may not be accurate Reset type: SYSRSn

16.9.2.2 CTRPHS Register (Offset = 2h) [Reset = 00000000h]

CTRPHS is shown in Figure 16-18 and described in Table 16-6.

Return to the Summary Table.

Counter Phase Offset Value Register

Figure 16-18. CTRPHS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRPHS																															
R/W-0h																															

Table 16-6. CTRPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CTRPHS	R/W	0h	Counter phase value register that can be programmed for phase lag/lead. This register CTRPHS is loaded into TSCTR upon either a SYNCI event or S/W force via a control bit. Used to achieve phase control synchronization with respect to other eCAP and PWM time-bases. This register is not applicable in HR mode. Reset type: SYSRSn

16.9.2.3 CAP1 Register (Offset = 4h) [Reset = 00000000h]

CAP1 is shown in Figure 16-19 and described in Table 16-7.

Return to the Summary Table.

Capture 1 Register

Figure 16-19. CAP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP1																															
R/W-0h																															

Table 16-7. CAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP1	R/W	0h	This register can be loaded (written) by: - Time-Stamp counter value (TSCTR) during a capture event - Software - may be useful for test purposes or initialization - ARPD shadow register (CAP3) when used in APWM mode Reset type: SYSRSn

16.9.2.4 CAP2 Register (Offset = 6h) [Reset = 00000000h]

CAP2 is shown in Figure 16-20 and described in Table 16-8.

Return to the Summary Table.

Capture 2 Register

Figure 16-20. CAP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP2																															
R/W-0h																															

Table 16-8. CAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP2	R/W	0h	This register can be loaded (written) by: - Time-Stamp (counter value) during a capture event - Software - may be useful for test purposes - ACMP shadow register (CAP4) when used in APWM mode Reset type: SYSRSn

16.9.2.5 CAP3 Register (Offset = 8h) [Reset = 00000000h]

CAP3 is shown in Figure 16-21 and described in Table 16-9.

Return to the Summary Table.

Capture 3 Register

Figure 16-21. CAP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP3																															
R/W-0h																															

Table 16-9. CAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP3	R/W	0h	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the period shadow (APRD) register. You can update the PWM period value through this register. CAP3 (APRD) shadows CAP1 in this mode. Reset type: SYSRSn

16.9.2.6 CAP4 Register (Offset = Ah) [Reset = 00000000h]

CAP4 is shown in Figure 16-22 and described in Table 16-10.

Return to the Summary Table.

Capture 4 Register

Figure 16-22. CAP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP4																															
R/W-0h																															

Table 16-10. CAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP4	R/W	0h	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the compare shadow (ACMP) register. You can update the PWM compare value via this register. CAP4 (ACMP) shadows CAP2 in this mode. Reset type: SYSRSn

16.9.2.7 ECCTL0 Register (Offset = 12h) [Reset = 0000007Fh]

ECCTL0 is shown in Figure 16-23 and described in Table 16-11.

Return to the Summary Table.

Capture Control Register 0

Figure 16-23. ECCTL0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									INPUTSEL						
R-0-0h									R/W-7Fh						

Table 16-11. ECCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6-0	INPUTSEL	R/W	7Fh	Capture input source select bits 0000000 capture input is ECAPxINPUT[0] 0000001 capture input is ECAPxINPUT[1] 0000010 capture input is ECAPxINPUT[2] ... 1111111 capture input is ECAPxINPUT[127] Reset type: CPU1.SYSRSn

16.9.2.8 ECCTL1 Register (Offset = 14h) [Reset = 0000h]

ECCTL1 is shown in Figure 16-24 and described in Table 16-12.

Return to the Summary Table.

Capture Control Register 1

Figure 16-24. ECCTL1 Register

15	14	13	12	11	10	9	8
FREE_SOFT		PRESCALE					CAPLDEN
R/W-0h		R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
CTRRST4	CAP4POL	CTRRST3	CAP3POL	CTRRST2	CAP2POL	CTRRST1	CAP1POL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 16-12. ECCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation Control Reset type: SYSRSn 0h (R/W) = TSCTR counter stops immediately on emulation suspend 1h (R/W) = TSCTR counter runs until = 0 2h (R/W) = TSCTR counter is unaffected by emulation suspend (Run Free) 3h (R/W) = TSCTR counter is unaffected by emulation suspend (Run Free)
13-9	PRESCALE	R/W	0h	Event Filter prescale select Reset type: SYSRSn 0h (R/W) = Divide by 1 (i.e,. no prescale, by-pass the prescaler) 1h (R/W) = Divide by 2 2h (R/W) = Divide by 4 3h (R/W) = Divide by 6 4h (R/W) = Divide by 8 5h (R/W) = Divide by 10 1Eh (R/W) = Divide by 60 1Fh (R/W) = Divide by 62
8	CAPLDEN	R/W	0h	Enable Loading of CAP1-4 registers on a capture event. Note that this bit does not disable CEVTn events from being generated. Reset type: SYSRSn 0h (R/W) = Disable CAP1-4 register loads at capture event time. 1h (R/W) = Enable CAP1-4 register loads at capture event time.
7	CTRRST4	R/W	0h	Counter Reset on Capture Event 4 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 4 (absolute time stamp operation) 1h (R/W) = Reset counter after Capture Event 4 time-stamp has been captured (used in difference mode operation)
6	CAP4POL	R/W	0h	Capture Event 4 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 4 triggered on a rising edge (RE) 1h (R/W) = Capture Event 4 triggered on a falling edge (FE)
5	CTRRST3	R/W	0h	Counter Reset on Capture Event 3 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 3 (absolute time stamp) 1h (R/W) = Reset counter after Event 3 time-stamp has been captured (used in difference mode operation)

Table 16-12. ECCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CAP3POL	R/W	0h	Capture Event 3 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 3 triggered on a rising edge (RE) 1h (R/W) = Capture Event 3 triggered on a falling edge (FE)
3	CTRRST2	R/W	0h	Counter Reset on Capture Event 2 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 2 (absolute time stamp) 1h (R/W) = Reset counter after Event 2 time-stamp has been captured (used in difference mode operation)
2	CAP2POL	R/W	0h	Capture Event 2 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 2 triggered on a rising edge (RE) 1h (R/W) = Capture Event 2 triggered on a falling edge (FE)
1	CTRRST1	R/W	0h	Counter Reset on Capture Event 1 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 1 (absolute time stamp) 1h (R/W) = Reset counter after Event 1 time-stamp has been captured (used in difference mode operation)
0	CAP1POL	R/W	0h	Capture Event 1 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 1 triggered on a rising edge (RE) 1h (R/W) = Capture Event 1 triggered on a falling edge (FE)

16.9.2.9 ECCTL2 Register (Offset = 15h) [Reset = 0006h]

ECCTL2 is shown in Figure 16-25 and described in Table 16-13.

Return to the Summary Table.

Capture Control Register 2

Figure 16-25. ECCTL2 Register

15	14	13	12	11	10	9	8
MODCNTRSTS		DMAEVTSEL		CTRFILTRESET	APWMPOL	CAP_APWM	SWSYNC
R-0h		R/W-0h		R-0/W1C-0h	R/W-0h	R/W-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
SYNCO_SEL		SYNCI_EN	TSCTRSTOP	REARM	STOP_WRAP		CONT_ONESHT
R/W-0h		R/W-0h	R/W-0h	R-0/W1S-0h	R/W-3h		R/W-0h

Table 16-13. ECCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	MODCNTRSTS	R	0h	This bit field reads current status on modulo counter 00b (R) = CAP1 register gets loaded on next capture event. 01b (R) = CAP2 register gets loaded on next capture event. 10b (R) = CAP3 register gets loaded on next capture event. 11b (R) = CAP4 register gets loaded on next capture event. Reset type: CPU1.SYSRSn
13-12	DMAEVTSEL	R/W	0h	DMA event select 00b (R/W) = DMA interrupt source is CEVT1 01b (R/W) = DMA interrupt source is CEVT2 10b (R/W) = DMA interrupt source is CEVT3 11b (R/W) = DMA interrupt source is CEVT4 Note: ECCTL1.CAPLDEN also needs to be set to '1' for ECAPxDMA_INT to be generated Reset type: CPU1.SYSRSn
11	CTRFILTRESET	R-0/W1C	0h	Reset Bit 0h (R) = No effect 1h (W) = Resets event filter, counter, modulo counter and CEVT[1,2,3,4] and CNTOVF , HRERROR flags Note: This provides an ability start capture module from known state in case spurious inputs are captured while ECAP is configured. Reset type: CPU1.SYSRSn
10	APWMPOL	R/W	0h	APWM output polarity select. This is applicable only in APWM operating mode. Reset type: SYSRSn 0h (R/W) = Output is active high (Compare value defines high time) 1h (R/W) = Output is active low (Compare value defines low time)
9	CAP_APWM	R/W	0h	CAP/APWM operating mode select Reset type: SYSRSn 0h (R/W) = ECAP module operates in capture mode. This mode forces the following configuration: - Inhibits TSCTR resets via CTR = PRD event - Inhibits shadow loads on CAP1 and 2 registers - Permits user to enable CAP1-4 register load - CAPx/APWMx pin operates as a capture input 1h (R/W) = ECAP module operates in APWM mode. This mode forces the following configuration: - Resets TSCTR on CTR = PRD event (period boundary - Permits shadow loading on CAP1 and 2 registers - Disables loading of time-stamps into CAP1-4 registers - CAPx/APWMx pin operates as a APWM output

Table 16-13. ECCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	SWSYNC	R-0/W1S	0h	Software-forced Counter (TSCTR) Synchronizer. This provides the user a method to generate a synchronization pulse through software. In APWM mode, the synchronization pulse can also be sourced from the CTR = PRD event. Reset type: SYSRSn 0h (R/W) = Writing a zero has no effect. Reading always returns a zero 1h (R/W) = Writing a one forces a TSCTR shadow load of current ECAP module and any ECAP modules down-stream providing the SYNCO_SEL bits are 0,0. After writing a 1, this bit returns to a zero.
7-6	SYNCO_SEL	R/W	0h	Sync-Out Select Reset type: SYSRSn 0h (R/W) = sync out signal is SWSYNC 1h (R/W) = Select CTR = PRD event to be the sync-out signal. Note: Selection CTR = PRD is meaningful only in APWM mode 2h (R/W) = Disable sync out signal 3h (R/W) = Disable sync out signal
5	SYNCI_EN	R/W	0h	Counter (TSCTR) Sync-In select mode Reset type: SYSRSn 0h (R/W) = Disable sync-in option 1h (R/W) = Enable counter (TSCTR) to be loaded from CTRPHS register upon either a SYNCI signal or a S/W force event.
4	TSCTRSTOP	R/W	0h	Time Stamp (TSCTR) Counter Stop (freeze) Control Reset type: SYSRSn 0h (R/W) = TSCTR stopped 1h (R/W) = TSCTR free-running
3	REARM	R-0/W1S	0h	Re-Arming Control. Note: The re-arm function is valid in one shot or continuous mode Reset type: SYSRSn 0h (R/W) = Has no effect (reading always returns a 0) 1h (R/W) = Arms the one-shot sequence as follows: 1) Resets the Mod4 counter to zero 2) Unfreezes the Mod4 counter 3) Enables capture register loads
2-1	STOP_WRAP	R/W	3h	Stop value for one-shot mode. This is the number (between 1-4) of captures allowed to occur before the CAP(1-4) registers are frozen, that is, capture sequence is stopped. Wrap value for continuous mode. This is the number (between 1-4) of the capture register in which the circular buffer wraps around and starts again. Notes: STOP_WRAP is compared to Mod4 counter and, when equal, 2 actions occur: - Mod4 counter is stopped (frozen) - Capture register loads are inhibited In one-shot mode, further interrupt events are blocked until re-armed. Reset type: SYSRSn 0h (R/W) = Stop after Capture Event 1 in one-shot mode Wrap after Capture Event 1 in continuous mode. 1h (R/W) = Stop after Capture Event 2 in one-shot mode Wrap after Capture Event 2 in continuous mode. 2h (R/W) = Stop after Capture Event 3 in one-shot mode Wrap after Capture Event 3 in continuous mode. 3h (R/W) = Stop after Capture Event 4 in one-shot mode Wrap after Capture Event 4 in continuous mode.
0	CONT_ONESHT	R/W	0h	Continuous or one-shot mode control (applicable only in capture mode) Reset type: SYSRSn 0h (R/W) = Operate in continuous mode 1h (R/W) = Operate in one-Shot mode

16.9.2.10 ECEINT Register (Offset = 16h) [Reset = 0000h]

ECEINT is shown in Figure 16-26 and described in Table 16-14.

Return to the Summary Table.

The interrupt enable bits (CEVT1, ...) block any of the selected events from generating an interrupt. Events will still be latched into the flag bit (ECFLG register) and can be forced/cleared via the ECFRC/ECCLR registers.
The proper procedure for configuring peripheral modes and interrupts is as follows:
- Disable global interrupts
- Stop eCAP counter
- Disable eCAP interrupts
- Configure peripheral registers
- Clear spurious eCAP interrupt flags
- Enable eCAP interrupts
- Start eCAP counter
- Enable global interrupts

Figure 16-26. ECEINT Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R/W-0h
7	6	5	4	3	2	1	0
CTR_EQ_CMP	CTR_EQ_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 16-14. ECEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	CTR_EQ_CMP	R/W	0h	Counter Equal Compare Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Compare Equal as an Interrupt source 1h (R/W) = Enable Compare Equal as an Interrupt source
6	CTR_EQ_PRD	R/W	0h	Counter Equal Period Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Period Equal as an Interrupt source 1h (R/W) = Enable Period Equal as an Interrupt source
5	CTROVF	R/W	0h	Counter Overflow Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disabled counter Overflow as an Interrupt source 1h (R/W) = Enable counter Overflow as an Interrupt source
4	CEVT4	R/W	0h	Capture Event 4 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 4 as an Interrupt source 1h (R/W) = Capture Event 4 Interrupt Enable
3	CEVT3	R/W	0h	Capture Event 3 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 3 as an Interrupt source 1h (R/W) = Enable Capture Event 3 as an Interrupt source
2	CEVT2	R/W	0h	Capture Event 2 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 2 as an Interrupt source 1h (R/W) = Enable Capture Event 2 as an Interrupt source

Table 16-14. ECEINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CEVT1	R/W	0h	Capture Event 1 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 1 as an Interrupt source 1h (R/W) = Enable Capture Event 1 as an Interrupt source
0	RESERVED	R	0h	Reserved

16.9.2.11 ECFLG Register (Offset = 17h) [Reset = 0000h]

ECFLG is shown in Figure 16-27 and described in Table 16-15.

Return to the Summary Table.

Capture Interrupt Flag Register

Figure 16-27. ECFLG Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R-0h
7	6	5	4	3	2	1	0
CTR_CMP	CTR_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-15. ECFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	CTR_CMP	R	0h	Compare Equal Compare Status Flag. This flag is active only in APWM mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) reached the compare register value (ACMP)
6	CTR_PRD	R	0h	Counter Equal Period Status Flag. This flag is only active in APWM mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) reached the period register value (APRD) and was reset.
5	CTROVF	R	0h	Counter Overflow Status Flag. This flag is active in CAP and APWM mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) has made the transition from FFFFFFFF to 00000000
4	CEVT4	R	0h	Capture Event 4 Status Flag This flag is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the fourth event occurred at ECAPx pin
3	CEVT3	R	0h	Capture Event 3 Status Flag. This flag is active only in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the third event occurred at ECAPx pin.
2	CEVT2	R	0h	Capture Event 2 Status Flag. This flag is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the second event occurred at ECAPx pin.
1	CEVT1	R	0h	Capture Event 1 Status Flag. This flag is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the first event occurred at ECAPx pin.

Table 16-15. ECFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	INT	R	0h	Global Interrupt Status Flag Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates that an interrupt was generated.

16.9.2.12 ECCLR Register (Offset = 18h) [Reset = 0000h]

ECCLR is shown in Figure 16-28 and described in Table 16-16.

Return to the Summary Table.

Capture Interrupt Clear Register

Figure 16-28. ECCLR Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CTR_CMP	CTR_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 16-16. ECCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	RESERVED	R-0/W1C	0h	Reserved
7	CTR_CMP	R-0/W1C	0h	Counter Equal Compare Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CTR=CMP flag.
6	CTR_PRD	R-0/W1C	0h	Counter Equal Period Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CTR=PRD flag.
5	CTROVF	R-0/W1C	0h	Counter Overflow Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CTROVF flag.
4	CEVT4	R-0/W1C	0h	Capture Event 4 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT4 flag.
3	CEVT3	R-0/W1C	0h	Capture Event 3 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT3 flag.
2	CEVT2	R-0/W1C	0h	Capture Event 2 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT2 flag.
1	CEVT1	R-0/W1C	0h	Capture Event 1 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT1 flag.
0	INT	R-0/W1C	0h	ECAP Global Interrupt Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the INT flag and enable further interrupts to be generated if any of the event flags are set to 1

16.9.2.13 ECFRC Register (Offset = 19h) [Reset = 0000h]

ECFRC is shown in Figure 16-29 and described in Table 16-17.

Return to the Summary Table.

Capture Interrupt Force Register

Figure 16-29. ECFRC Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CTR_CMP	CTR_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 16-17. ECFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	RESERVED	R-0/W1S	0h	Reserved
7	CTR_CMP	R-0/W1S	0h	Force Counter Equal Compare Interrupt. This event is only active in APWM mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CTR_CMP flag.
6	CTR_PRD	R-0/W1S	0h	Force Counter Equal Period Interrupt. This event is only active in APWM mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CTR_PRD flag.
5	CTROVF	R-0/W1S	0h	Force Counter Overflow Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 to this bit sets the CTROVF flag.
4	CEVT4	R-0/W1S	0h	Force Capture Event 4. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT4 flag.
3	CEVT3	R-0/W1S	0h	Force Capture Event 3. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT3 flag.
2	CEVT2	R-0/W1S	0h	Force Capture Event 2. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT2 flag.
1	CEVT1	R-0/W1S	0h	Force Capture Event 1. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Sets the CEVT1 flag.
0	RESERVED	R	0h	Reserved

16.9.2.14 ECAPSYNCINSEL Register (Offset = 1Eh) [Reset = 00000001h]

ECAPSYNCINSEL is shown in Figure 16-30 and described in Table 16-18.

Return to the Summary Table.

SYNC source select register

Figure 16-30. ECAPSYNCINSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											SEL				
R-0h																											R/W-1h				

Table 16-18. ECAPSYNCINSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-0	SEL	R/W	1h	These bits determines the source of SYNCIN signal. 0x0 : Disabled using SOC tieoff. 0x1-0x7F : Refer to MCPWM SYNCIN sources in MCPWM TRM chapter. Reset type: SYSRSn

Chapter 17

Enhanced Quadrature Encoder Pulse (eQEP)

The enhanced Quadrature Encoder Pulse (eQEP) module described here is a Type 2 eQEP. See the *C2000 Real-Time Control Peripheral Reference Guide* for a list of all devices with a module of the same type to determine the differences between types and for a list of device-specific differences within a type.

The enhanced quadrature encoder pulse (eQEP) module is used for direct interface with a linear or rotary incremental encoder to get position, direction, and speed information from a rotating machine for use in a high-performance motion and position-control system.

17.1 Introduction

An incremental encoder disk is patterned with a track of slots along the periphery, as shown in Figure 17-1. These slots create an alternating pattern of dark and light lines. The disk count is defined as the number of dark and light line pairs that occur per revolution (lines per revolution). As a rule, a second track is added to generate a signal that occurs once per revolution (index signal: QEPI), which can be used to indicate an absolute position. Encoder manufacturers identify the index pulse using different terms such as index, marker, home position, and zero reference

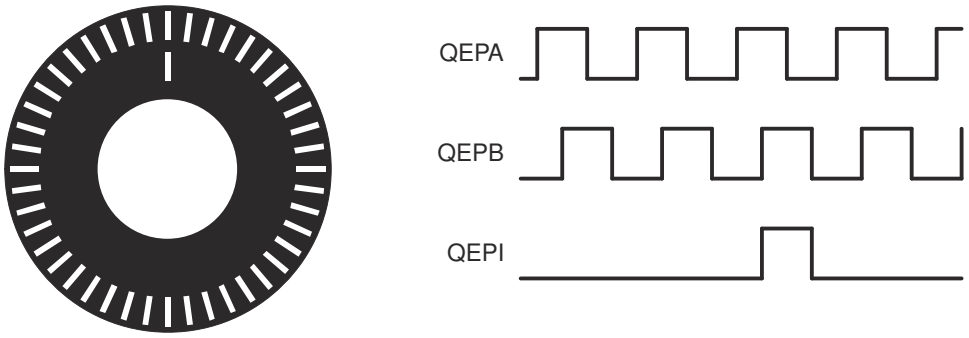
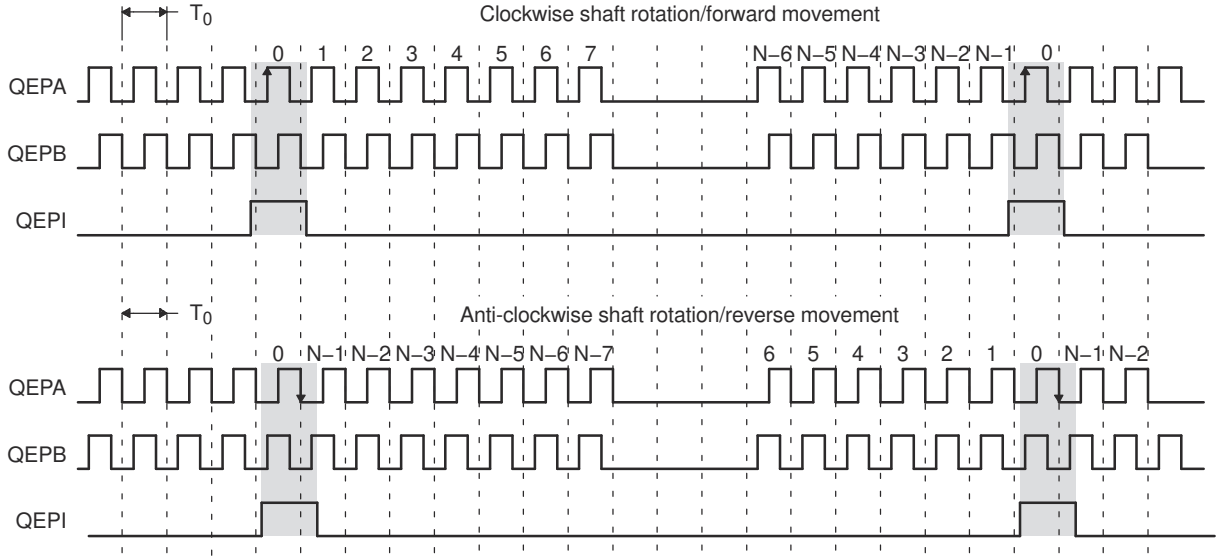


Figure 17-1. Optical Encoder Disk

To derive direction information, the lines on the disk are read out by two different photo-elements that "look" at the disk pattern with a mechanical shift of 1/4 the pitch of a line pair between them. This shift is detected with a reticle or mask that restricts the view of the photo-element to the desired part of the disk lines. As the disk rotates, the two photo-elements generate signals that are shifted 90° out of phase from each other. These are commonly called the quadrature QEPA and QEPB signals. The clockwise direction for most encoders is defined as the QEPA channel going positive before the QEPB channel and conversely, as shown in Figure 17-2.



Legend: N = lines per revolution

Figure 17-2. QEP Encoder Output Signal for Forward/Reverse Movement

The encoder wheel typically makes one revolution for every revolution of the motor, or the wheel can be at a geared rotation ratio with respect to the motor. Therefore, the frequency of the digital signal coming from the QEPA and QEPB outputs varies proportionally with the velocity of the motor. For example, a 2000-line encoder

directly coupled to a motor running at 5000 revolutions-per-minute (rpm) results in a frequency of 166.6kHz, so by measuring the frequency of either the QEPA or QEPB output, the processor can determine the velocity of the motor.

Quadrature encoders from different manufacturers come with two forms of index pulse (gated index pulse or ungated index pulse) as shown in Figure 17-3. A nonstandard form of index pulse is ungated. In the ungated configuration, the index edges are not necessarily coincident with A and B signals. The gated index pulse is aligned to any of the four quadrature edges and width of the index pulse and can be equal to a quarter, half, or full period of the quadrature signal.

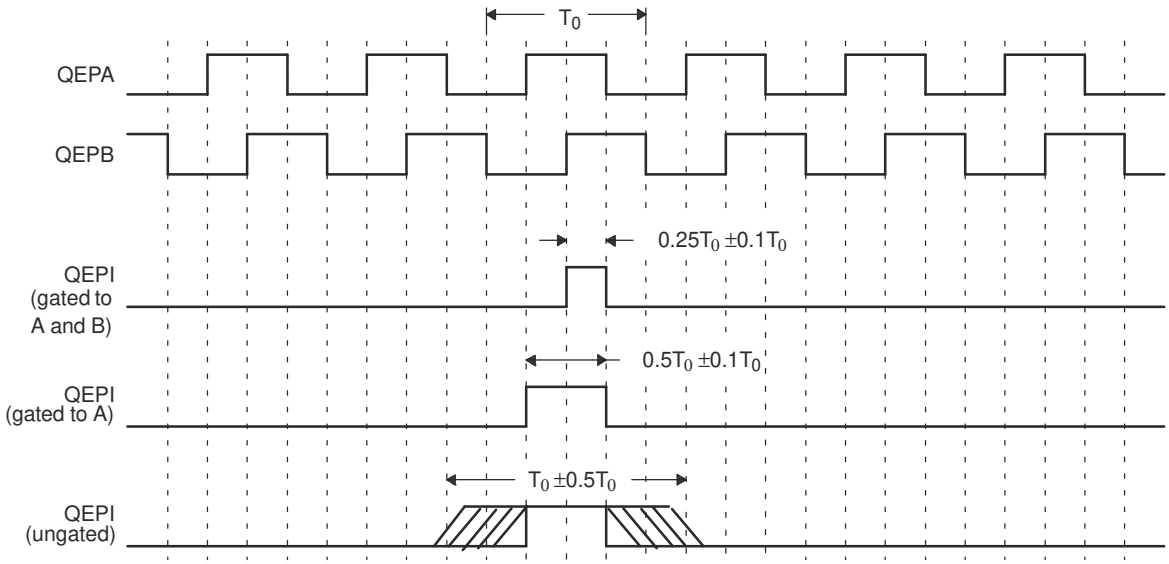


Figure 17-3. Index Pulse Example

Some typical applications of shaft encoders include robotics and computer input in the form of a mouse. Inside your mouse you can see where the mouse ball spins a pair of axles (a left/right, and an up/down axle). These axles are connected to optical shaft encoders that effectively tell the computer how fast and in what direction the mouse is moving.

General Issues: Estimating velocity from a digital position sensor is a cost-effective strategy in motor control. Two different first order approximations for velocity can be written as:

$$v(k) \approx \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \quad (21)$$

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (22)$$

where:

- v(k) = Velocity at time instant k
- x(k) = Position at time instant k
- x(k-1) = Position at time instant k-1
- T = Fixed unit time or inverse of velocity calculation rate
- ΔX = Incremental position movement in unit time
- t(k) = Time instant "k"
- t(k-1) = Time instant "k-1"
- X = Fixed unit position
- ΔT = Incremental time elapsed for unit position movement

Equation 21 is the conventional approach to velocity estimation and requires a time base to provide a unit time event for velocity calculation. Unit time is basically the inverse of the velocity calculation rate.

The encoder count (position) is read once during each unit time event. The quantity [x(k) - x(k-1)] is formed by subtracting the previous reading from the current reading. Then the velocity estimate is computed by multiplying by the known constant 1/T (where T is the constant time between unit time events and is known in advance).

Estimation based on Equation 21 has an inherent accuracy limit directly related to the resolution of the position sensor and the unit time period T. For example, consider a 500 line-per-revolution quadrature encoder with a velocity calculation rate of 400Hz. When used for position, the quadrature encoder gives a four-fold increase in resolution; in this case, 2000 counts-per-revolution. The minimum rotation that can be detected is, therefore, 0.0005 revolutions, which gives a velocity resolution of 12rpm when sampled at 400Hz. While this resolution can be satisfactory at moderate or high speeds, for example 1% error at 1200rpm, this resolution clearly proves inadequate at low speeds. In fact, at speeds below 12rpm, the speed estimate is erroneously zero much of the time.

At low speed, Equation 22 provides a more accurate approach. This method requires a position sensor that outputs a fixed interval pulse train, such as the aforementioned quadrature encoder. The width of each pulse is defined by motor speed for a given sensor resolution. Equation 22 can be used to calculate motor speed by measuring the elapsed time between successive quadrature pulse edges. However, this method suffers from the opposite limitation, as does Equation 21. A combination of relatively large motor speeds and high sensor resolution makes the time interval ΔT small, and thus more greatly influenced by the timer resolution. This can introduce considerable error into high-speed estimates.

For systems with a large speed range (that is, speed estimation is needed at both low and high speeds), one approach is to use Equation 22 at low speed and have the DSP software switch over to Equation 21 when the motor speed rises above some specified threshold.

17.1.1 EQEP Related Collateral

Foundational Materials

- C28x Academy - EQEP
- Interfacing with Quadrature Encoders (Video)
- *Real-Time Control Reference Guide*
  - Refer to the Encoders section

Getting Started Materials

- *C2000™ Position Manager PTO API Reference Guide Application Report*

Expert Materials

- *CW/CCW Support on the C2000 eQEP Module Application Report*

17.2 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

For proper operation of the eQEP module, input GPIO pins must be configured using the GPxQSELn registers for synchronous input mode (with or without qualification). The asynchronous mode cannot be used for eQEP input pins. The internal pullups can be configured in the GPyPUD register.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

17.3 Description

This section provides the eQEP inputs, memory map, and functional description.

17.3.1 EQEP Inputs

The eQEP inputs include two pins for quadrature-clock mode or direction-count mode, an index (or 0 marker), and a strobe input. The eQEP module requires that the QEPA, QEPB, and QEPI inputs are synchronized to SYSCLK prior to entering the module.

The application code can enable the synchronous GPIO input feature on any eQEP-enabled GPIO pins (see the *General-Purpose Input/Output (GPIO)* chapter for more details).

- **QEPA/XCLK and QEPB/XDIR:** These two pins can be used in quadrature-clock mode or direction-count mode.
  - Quadrature-clock Mode: The eQEP encoders provide two square wave signals (A and B) 90 electrical degrees out of phase. This phase relationship is used to determine the direction of rotation of the input shaft and number of eQEP pulses from the index position to derive the relative position information. For forward or clockwise rotation, QEPA signal leads QEPB signal and conversely. The quadrature decoder uses these two inputs to generate quadrature-clock and direction signals.
  - Direction-count Mode: In direction-count mode, direction and clock signals are provided directly from the external source. Some position encoders have this type of output instead of quadrature output. The QEPA pin provides the clock input and the QEPB pin provides the direction input.
- **QEPI: Index or Zero Marker:** The eQEP encoder uses an index signal to assign an absolute start position from which position information is incrementally encoded using quadrature pulses. This pin is connected to the index output of the eQEP encoder to optionally reset the position counter for each revolution. This signal can be used to initialize or latch the position counter on the occurrence of a desired event on the index pin.
- **QEPS: Strobe Input:** This general-purpose strobe signal can initialize or latch the position counter on the occurrence of a desired event on the strobe pin. This signal is typically connected to a sensor or limit switch to notify that the motor has reached a defined position.

Input signals to the eQEP (QEPA, QEPB, QEPI and QEPS) can come from multiple sources; that is, device pin, CMPSSx, Input X-BAR, or PWMXBARx. One typical use case is if SinCos transducers are used in the motor control system to estimate the position of motor shaft and Index signal is coming from traditional rotary encoder, source of the eQEP signals (QEPA, QEPB and QEPI) can be configured as output of CMPSSx that decodes the Sin, Cos, and Index signals. Figure 17-4 illustrates this.

Selection of the source of Input signals (QEPA, QEPB, and QEPI) is user-configurable through the QEPSRCSEL register as shown in Table 17-1.

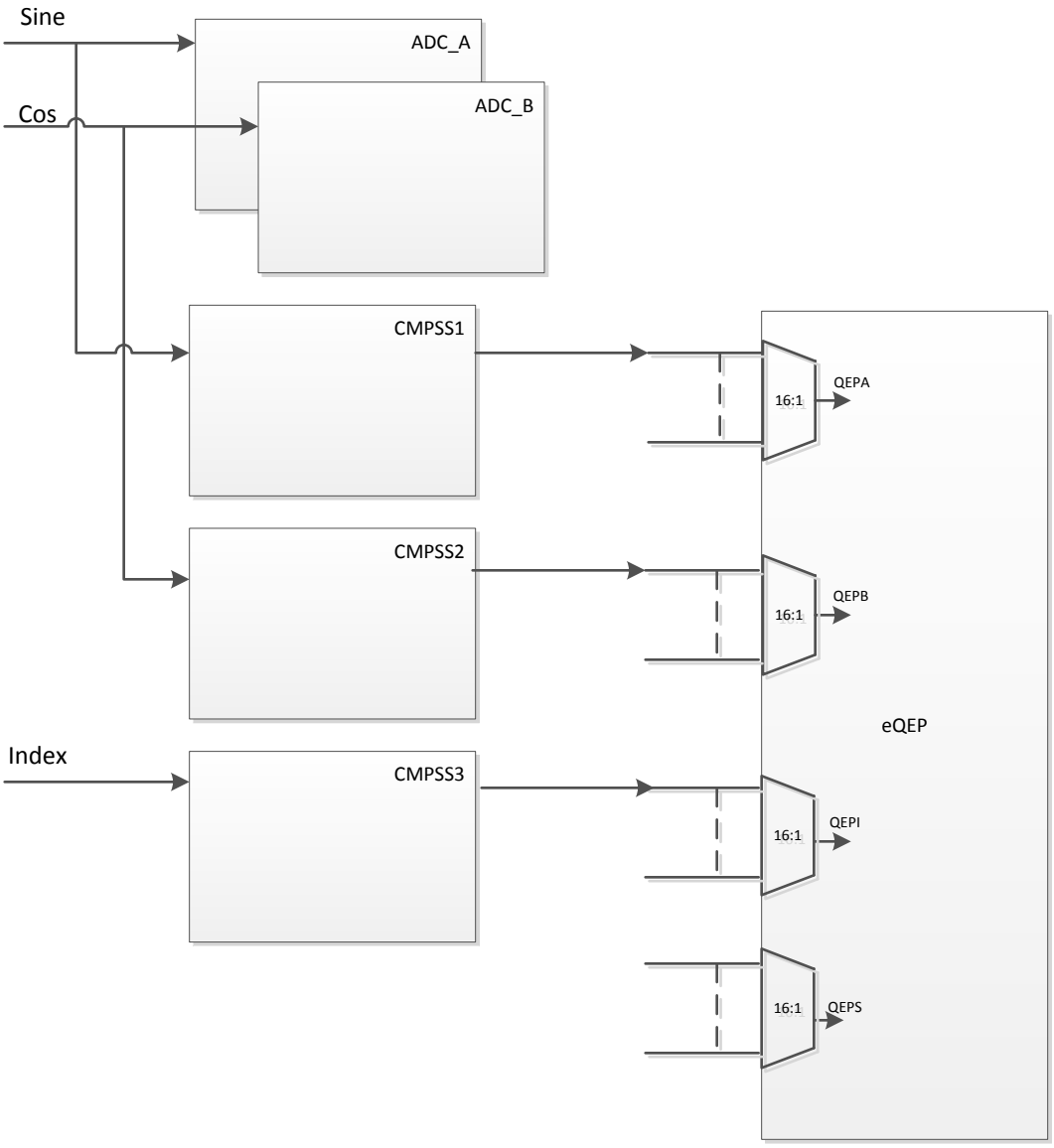


Figure 17-4. Using eQEP to Decode Signals from SinCos Transducer

Table 17-1. eQEP Input Source Select Table

QEPASEL, QEPBSEL, QEPISEL	Input Signal
0	Device Pin
1	CMPSS1_CTRIPH
2	CMPSS2_CTRIPH
3	CMPSS3_CTRIPH
4	
5	Tie zero
6	PWMXBAR1
7	PWMXBAR2
8	PWMXBAR3
9	PWMXBAR4
10	PWMXBAR5
11	PWMXBAR6
12	PWMXBAR7
13	PWMXBAR8
14	Reserved
15	Reserved

Note

Configuration of QEPSRCSEL register to select the source of QEPA, QEPB, and QEPI signals can lead to unexpected transition on these signals, which can cause an undesirable outcome if eQEP is already running. Make sure that the eQEP is disabled before configuring the QEPSRCSEL register for input signals.

17.3.2 Functional Description

The eQEP peripheral contains the following major functional units (as shown in Figure 17-5):

- Programmable input qualification for each pin (part of the GPIO MUX)
- Quadrature decoder unit (QDU)
- Position counter and control unit for position measurement (PCCU)
- Quadrature edge-capture unit for low-speed measurement (QCAP)
- Unit time base for speed/frequency measurement (UTIME)
- Watchdog timer for detecting stalls (QWDOG)
- Quadrature Mode Adapter (QMA)

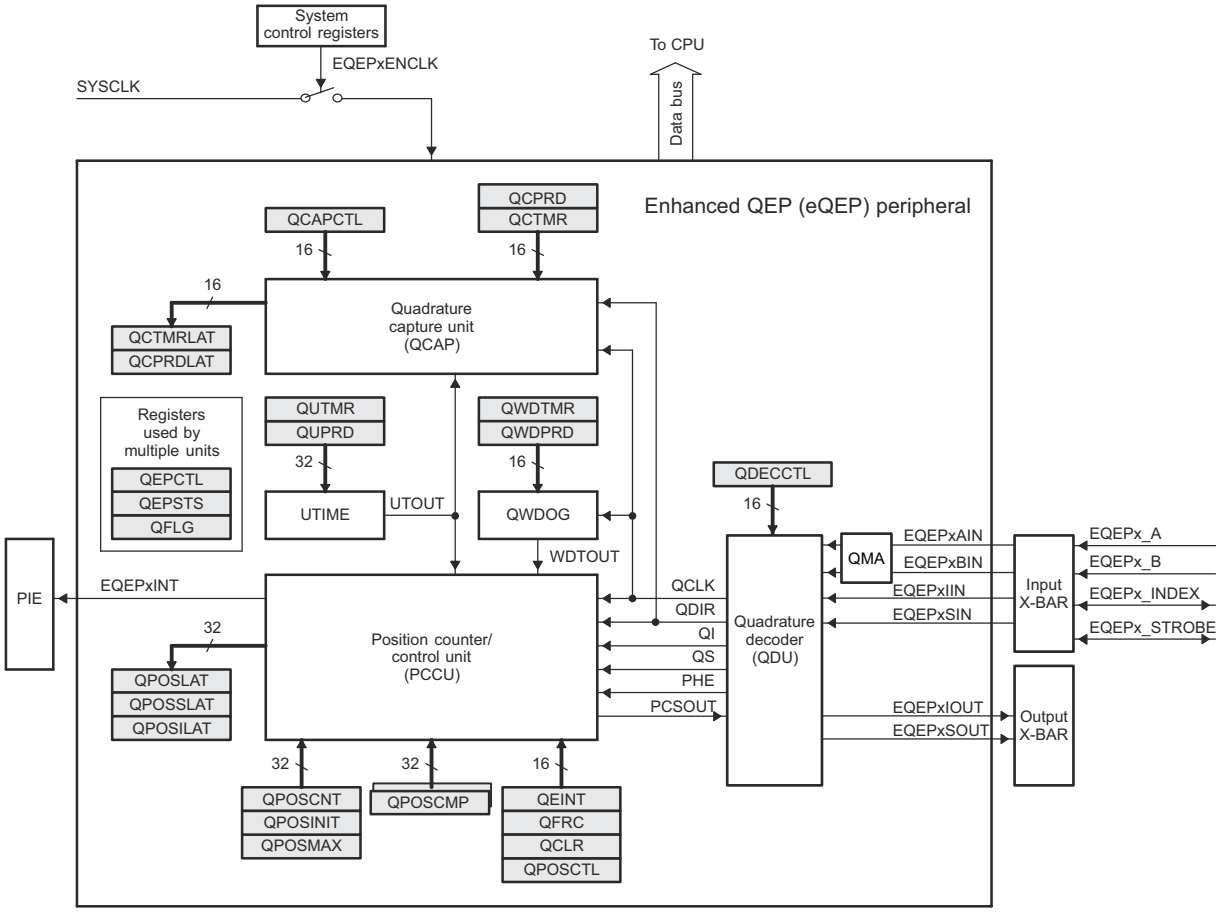


Figure 17-5. Functional Block Diagram of the eQEP Peripheral

17.3.3 eQEP Memory Map

Table 17-2 lists the registers with the memory locations, sizes, and reset values.

Table 17-2. EQEP Memory Map

Name	Offset	Size(x16)/ #shadow	Reset	Register Description
QPOSCNT	0x00	2/0	0x0000 0000	eQEP Position Counter
QPOSINIT	0x02	2/0	0x0000 0000	eQEP Initialization Position Count
QPOSMAX	0x04	2/0	0x0000 0000	eQEP Maximum Position Count
QPOSCMP	0x06	2/1	0x0000 0000	eQEP Position-compare
QPOSILAT	0x08	2/0	0x0000 0000	eQEP Index Position Latch
QPOSSLAT	0x0A	2/0	0x0000 0000	eQEP Strobe Position Latch
QPOSLAT	0x0C	2/0	0x0000 0000	eQEP Position Latch
QUTMR	0x0E	2/0	0x0000 0000	eQEP Unit Timer
QUPRD	0x10	2/0	0x0000 0000	eQEP Unit Period Register
QWDTMR	0x12	1/0	0x0000	eQEP Watchdog Timer
QWDPRD	0x13	1/0	0x0000	eQEP Watchdog Period Register
QDECCTL	0x14	1/0	0x0000	eQEP Decoder Control Register
QEPCTL	0x15	1/0	0x0000	eQEP Control Register
QCAPCTL	0x16	1/0	0x0000	eQEP Capture Control Register
QPOSCTL	0x17	1/0	0x0000	eQEP Position-compare Control Register
QEINT	0x18	1/0	0x0000	eQEP Interrupt Enable Register
QFLG	0x19	1/0	0x0000	eQEP Interrupt Flag Register
QCLR	0x1A	1/0	0x0000	eQEP Interrupt Clear Register
QFRC	0x1B	1/0	0x0000	eQEP Interrupt Force Register
QEPSTS	0x1C	1/0	0x0000	eQEP Status Register
QCTMR	0x1D	1/0	0x0000	eQEP Capture Timer
QCPRD	0x1E	1/0	0x0000	eQEP Capture Period Register
QCTMRLAT	0x1F	1/0	0x0000	eQEP Capture Timer Latch
QCPRDLAT	0x20	1/0	0x0000	eQEP Capture Period Latch
Reserved	0x21 to 0x2F	15/0		
REV	0x30	2/0	0x0000	eQEP Revision Number
QEPSTROBESEL	0x32	2/0	0x0000	eQEP Strobe select register
QMACTRL	0x34	2/0	0x0000	eQEP QMA Control register
QEPSRCSEL	0x36	2/0	0x0000	eQEP Source Select Register
Reserved	0x38 to 0x3F	8/0		

17.4 Quadrature Decoder Unit (QDU)

Figure 17-6 shows a functional block diagram of the QDU.

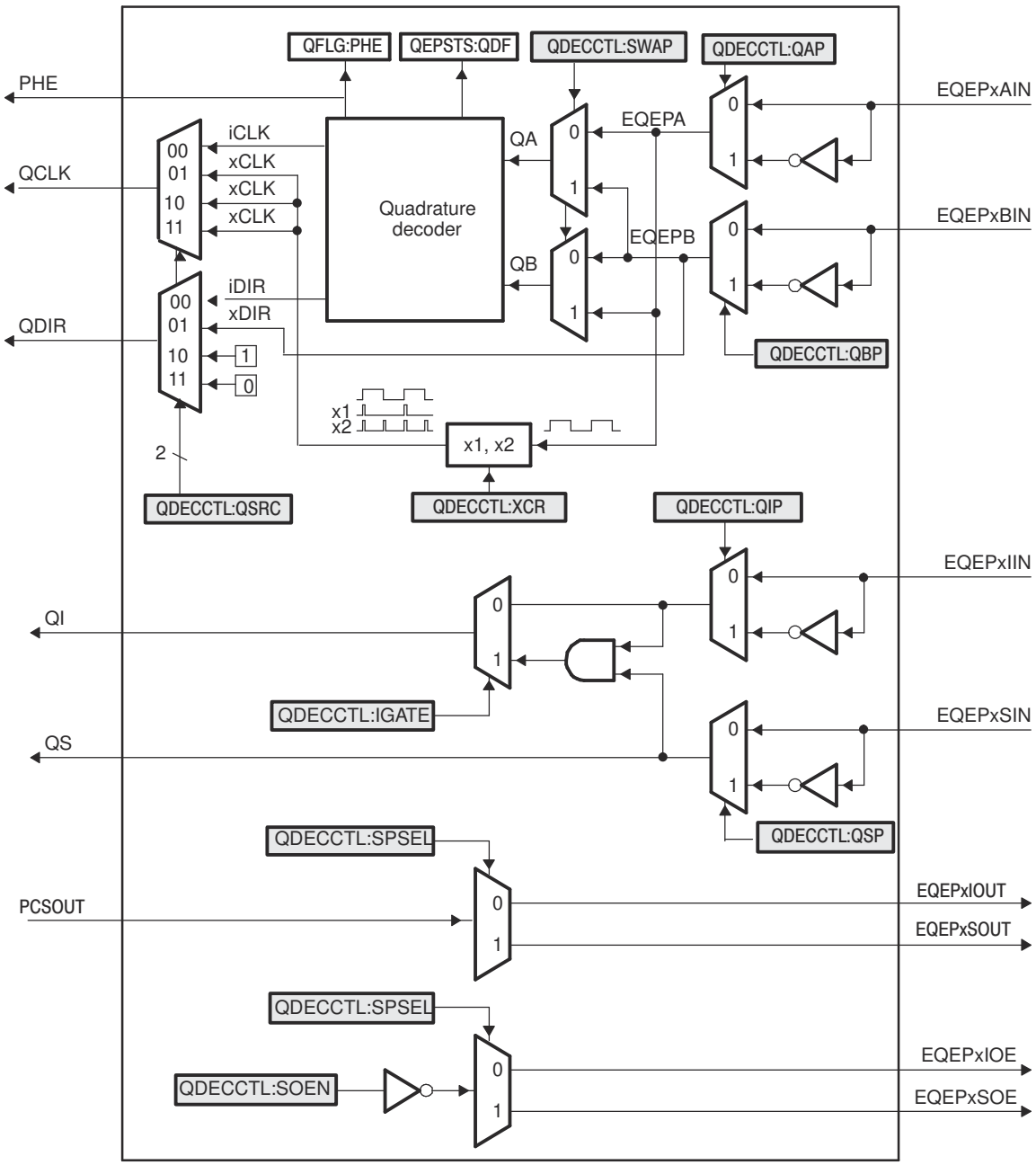


Figure 17-6. Functional Block Diagram of Decoder Unit

17.4.1 Position Counter Input Modes

Clock and direction input to the position counter is selected using QDECCTL[QSRC] bits, based on interface input requirement as follows:

- Quadrature-count mode
- Direction-count mode
- UP-count mode
- DOWN-count mode

17.4.1.1 Quadrature Count Mode

The quadrature decoder generates the direction and clock to the position counter in quadrature count mode.

Direction Decoding

The direction decoding logic of the eQEP circuit determines which one of the sequences (QEPA, QEPB) is the leading sequence and accordingly updates the direction information in the QEPSTS[QDF] bit. Table 17-3 and Figure 17-7 show the direction decoding logic in truth table and state machine form. Both edges of the QEPA and QEPB signals are sensed to generate count pulses for the position counter. Therefore, the frequency of the clock generated by the eQEP logic is four times that of each input sequence. Figure 17-8 shows the direction decoding and clock generation from the eQEP input signals.

Table 17-3. Quadrature Decoder Truth Table

Previous Edge	Present Edge	QDIR	QPOSCNT
QA↑	QB↑	UP	Increment
	QB↓	DOWN	Decrement
	QA↓	TOGGLE	Increment or Decrement
QA↓	QB↓	UP	Increment
	QB↑	DOWN	Decrement
	QA↑	TOGGLE	Increment or Decrement
QB↑	QA↑	DOWN	Decrement
	QA↓	UP	Increment
	QB↓	TOGGLE	Increment or Decrement
QB↓	QA↓	DOWN	Decrement
	QA↑	UP	Increment
	QB↑	TOGGLE	Increment or Decrement

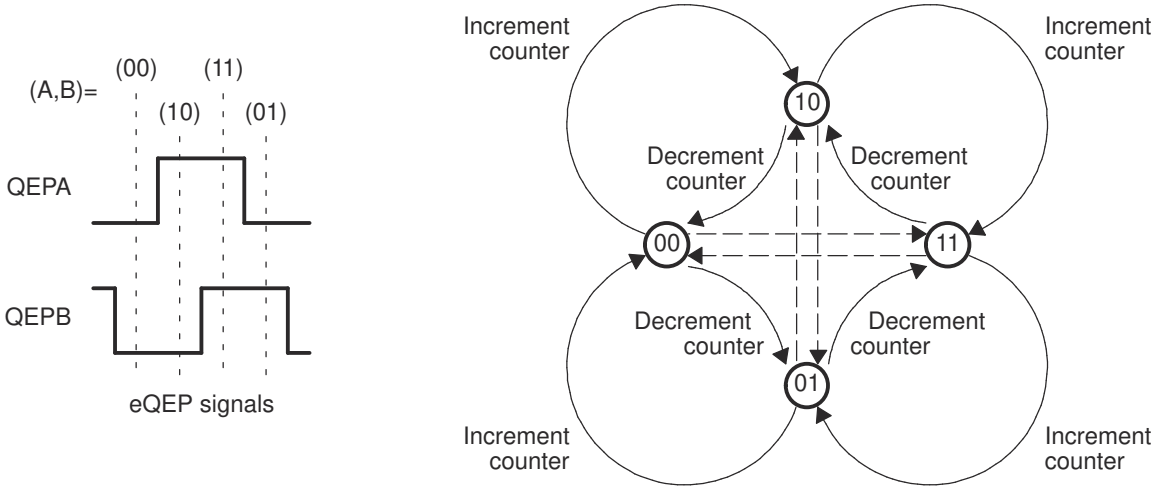


Figure 17-7. Quadrature Decoder State Machine

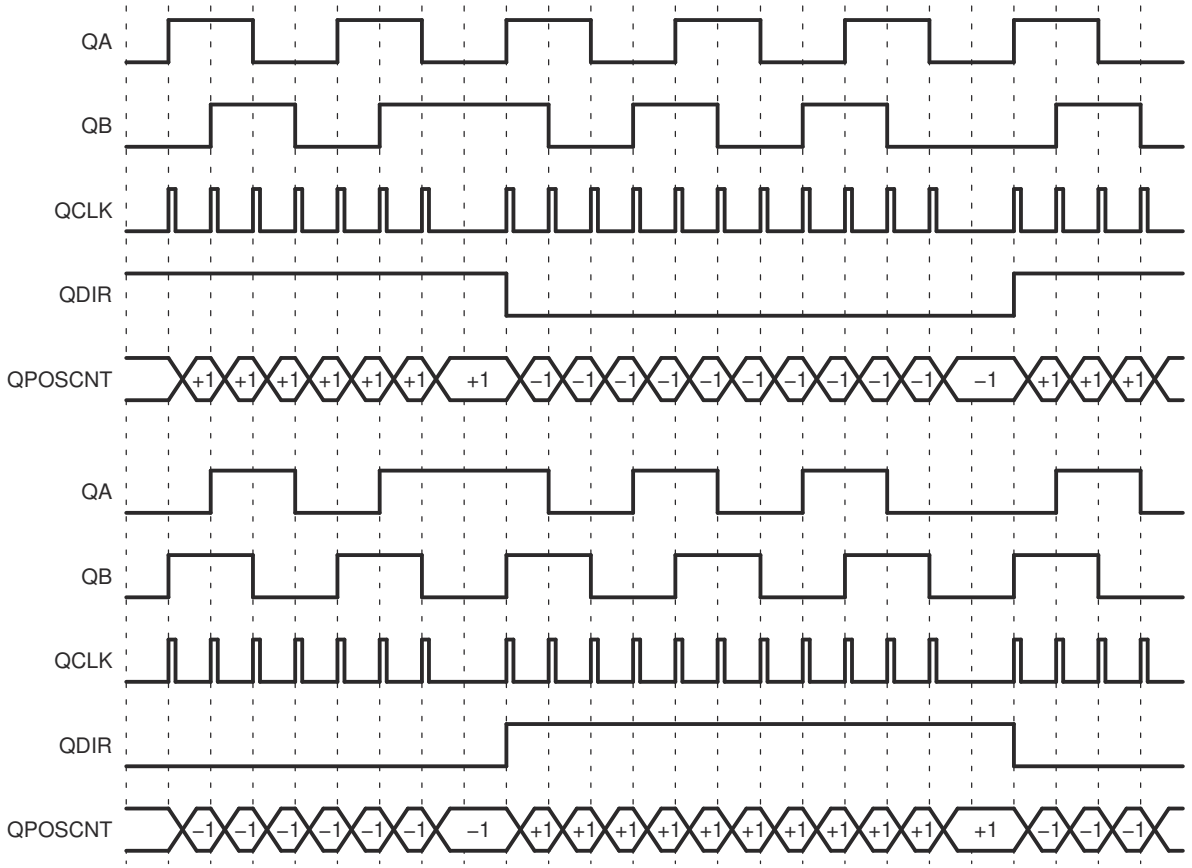


Figure 17-8. Quadrature-clock and Direction Decoding

Phase Error Flag In normal operating conditions, quadrature inputs QEPA and QEPB is 90 degrees out of phase. The phase error flag (PHE) is set in the QFLG register and the QPOSCNT value can be incorrect and offset by multiples of 1 or 3. That is, when edge transition is detected simultaneously on the QEPA and QEPB signals to optionally generate interrupts. State transitions marked by dashed lines in Figure 17-7 are invalid transitions that generate a phase error.

Count Multiplication The eQEP position counter provides 4x times the resolution of an input clock by generating a quadrature-clock (QCLK) on the rising/falling edges of both eQEP input clocks (QEPA and QEPB) as shown in Figure 17-8.

Reverse Count In normal quadrature count operation, QEPA input is applied to the QA input of the quadrature decoder and the QEPB input is applied to the QB input of the quadrature decoder. Reverse counting is enabled by setting the SWAP bit in the QDECCTL register. This swaps the input to the quadrature decoder; thereby, reversing the counting direction.

17.4.1.2 Direction-Count Mode

Some position encoders provide direction and clock outputs, instead of quadrature outputs. In such cases, direction-count mode can be used. The QEPA input provides the clock for the position counter and the QEPB input has the direction information. The position counter is incremented on every rising edge of a QEPA input when the direction input is high, and decremented when the direction input is low.

17.4.1.3 Up-Count Mode

The counter direction signal is hard-wired for up-count and the position counter is used to measure the frequency of the QEPA input. Clearing the QDECCTL[XCR] bit enables clock generation to the position counter on both edges of the QEPA input; thereby, increasing the measurement resolution by a factor of 2x. In up-count mode, we recommend that the application not configure QEPB as a GPIO mux option, or make sure that a signal edge is not generated on the QEPB input.

17.4.1.4 Down-Count Mode

The counter direction signal is hardwired for a down-count and the position counter is used to measure the frequency of the QEPA input. Clearing the QDECCTL[XCR] bit enables clock generation to the position counter on both edges of a QEPA input, thereby increasing the measurement resolution by a factor of 2x. In down-count mode, the application must not configure QEPB as a GPIO mux option or make sure that a signal edge is not generated on the QEPB input.

17.4.2 eQEP Input Polarity Selection

Each eQEP input can be inverted using QDECCTL[8:5] control bits. As an example, setting the QDECCTL[QIP] bit inverts the index input.

17.4.3 Position-Compare Sync Output

The enhanced eQEP peripheral includes a position-compare unit that is used to generate the position-compare sync signal on compare match between the position-counter register (QPOSCNT) and the position- compare register (QPOSCMP). This sync signal can be output using an index pin or strobe pin of the EQEP peripheral.

Setting the QDECCTL[SOEN] bit enables the position-compare sync output and the QDECCTL[SPSEL] bit selects either an eQEP index pin or an eQEP strobe pin.

17.5 Position Counter and Control Unit (PCCU)

The position-counter and control unit provides two configuration registers (QEPCTL and QPOSCTL) for setting up position-counter operational modes, position-counter initialization/latch modes and position-compare logic for sync signal generation.

17.5.1 Position Counter Operating Modes

Position-counter data can be captured in different manners. In some systems, the position counter is accumulated continuously for multiple revolutions and the position-counter value provides the position information with respect to the known reference. An example of this is the quadrature encoder mounted on the motor controlling the print head in the printer. Here the position counter is reset by moving the print head to the home position and then the position counter provides absolute position information with respect to home position.

In other systems, the position counter is reset on every revolution using index pulse, and the position counter provides a rotor angle with respect to the index pulse position.

The position counter can be configured to operate in following four modes

- Position-Counter Reset on Index Event
- Position-Counter Reset on Maximum Position
- Position-Counter Reset on the first Index Event
- Position-Counter Reset on Unit Time Out Event (Frequency Measurement)

In all the above operating modes, the position counter is reset to 0 on overflow and to the QPOSMAX register value on underflow. Overflow occurs when the position counter counts up after the QPOSMAX value. Underflow occurs when the position counter counts down after 0. The Interrupt flag is set to indicate overflow/underflow in QFLG register.

17.5.1.1 Position Counter Reset on Index Event (QEPCTL[PCRM] = 00)

If the index event occurs during the forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock.

The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in QEPSTS registers, the eQEP peripheral also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for index event reset operation.

For example, if the first reset operation occurs on the falling edge of QEPB during the forward direction, then all the subsequent reset must be aligned with the falling edge of QEPB for the forward rotation and on the rising edge of QEPB for the reverse rotation as shown in Figure 17-9.

The position-counter value is latched to the QPOSILAT register and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker. The position-counter error flag (QEPSTS[PCEF]) and error interrupt flag (QFLG[PCE]) are set if the latched value is not equal to 0 or QPOSMAX. The position-counter error flag (QEPSTS[PCEF]) is updated on every index event marker and an interrupt flag (QFLG[PCE]) is set on error that can be cleared only through software.

The index event latch configuration QEPCTL[IEL] must be configured to 00 or 11 when PRCM = 0 and the position counter error flag/interrupt flag are generated only in index event reset mode. The position counter value is latched into the IPOSLAT register on every index marker.

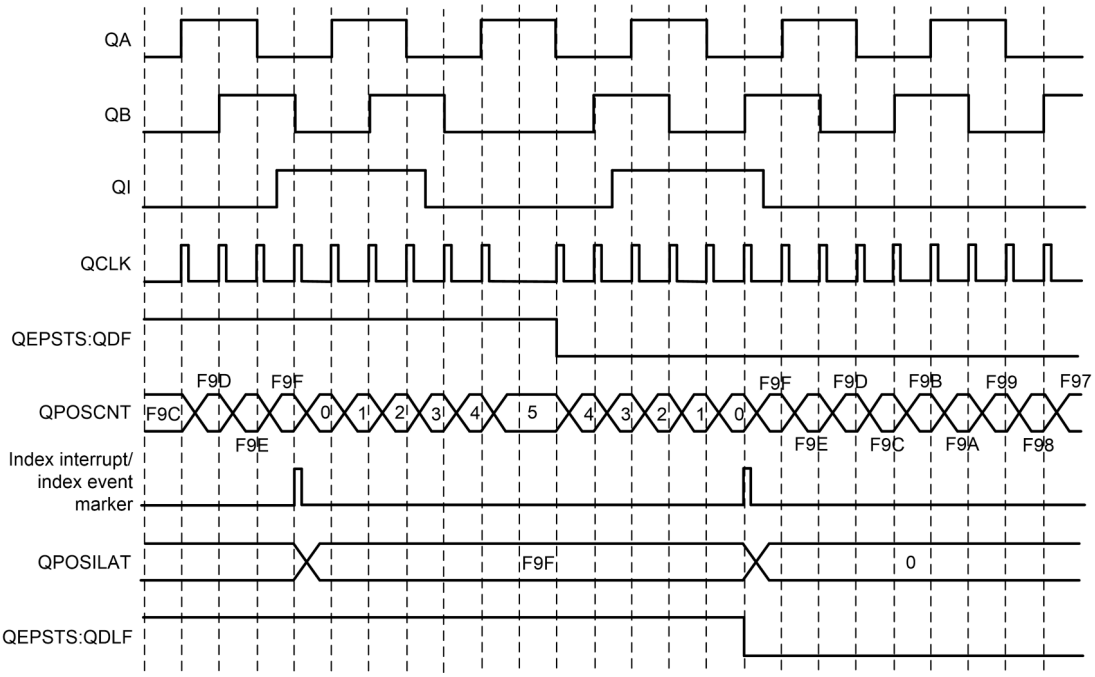


Figure 17-9. Position Counter Reset by Index Pulse for 1000-Line Encoder (QPOSMAX = 3999 or 0xF9F)

Note

In case of a boundary condition where the time period between the Index Event and the previous QCLK edge is less than SYSCLK period, then QPOSCNT gets reset to zero or QPOSMAX in the same SYSCLK cycle and does not wait for the next QCLK edge to occur.

17.5.1.2 Position Counter Reset on Maximum Position (QEPCTL[PCRM] = 01)

If the position counter is equal to QPOSMAX, then the position counter is reset to 0 on the next eQEP clock for forward movement and position counter overflow flag is set. If the position counter is equal to ZERO, then the position counter is reset to QPOSMAX on the next QEP clock for reverse movement and position-counter underflow flag is set. Figure 17-10 shows the position-counter reset operation in this mode.

The first index marker fields (QEPSTS[FIDF] and QEPSTS[FIMF]) are not applicable in this mode.

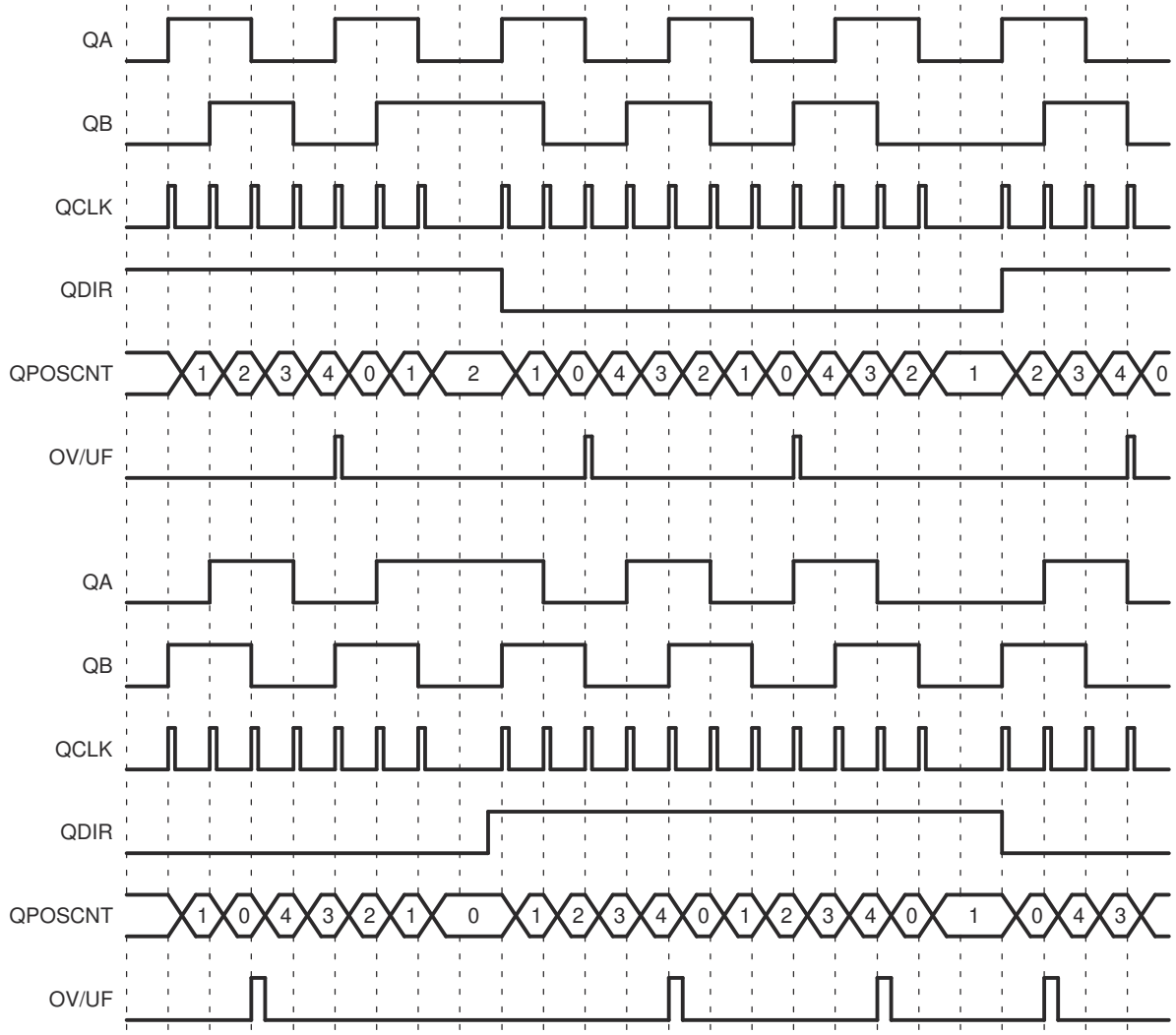


Figure 17-10. Position Counter Underflow/Overflow (QPOSMAX = 4)

17.5.1.3 Position Counter Reset on the First Index Event (QEPCTL[PCRM] = 10)

If the index event occurs during forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock. Note that this is done only on the first occurrence and subsequently the position-counter value is not reset on an index event; rather, the position-counter value is reset based on the maximum position as described in Section 17.5.1.2.

The first index marker fields (QEPSTS[FIDF] and QEPSTS[FIMF]) are not applicable in this mode.

17.5.1.4 Position Counter Reset on Unit Time-out Event (QEPCTL[PCRM] = 11)

In this mode, QPOSCNT is set to 0 or QPOMAX, depending on the direction mode selected by QDECCTL[QSRC] bits on a unit time event. This is useful for frequency measurement.

17.5.2 Position Counter Latch

The eQEP index and strobe input can be configured to latch the position counter (QPOSCNT) into QPOSILAT and QPOSSLAT, respectively, on occurrence of a definite event on these pins.

17.5.2.1 Index Event Latch

In some applications, it is not desirable to reset the position counter on every index event and instead it can be required to operate the position counter in full 32-bit mode (QEPCTL[PCRM] = 01 and QEPCTL[PCRM] = 10 modes).

In such cases, the eQEP position counter can be configured to latch on the following events and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker.

- Latch on Rising edge (QEPCTL[IEL] = 01)
- Latch on Falling edge (QEPCTL[IEL] = 10)
- Latch on Index Event Marker (QEPCTL[IEL] = 11)

This is particularly useful as an error checking mechanism to check if the position counter accumulated the correct number of counts between index events. As an example, the 1000-line encoder must count 4000 times when moving in the same direction between the index events.

The index event latch interrupt flag (QFLG[IEL]) is set when the position counter is latched to the QPOSILAT register. The index event latch configuration bits (QEPCTL[IEL]) are ignored when QEPCTL[PCRM] = 00.

Latch on Rising Edge (QEPCTL[IEL] = 01)	The position-counter value (QPOSCNT) is latched to the QPOSILAT register on every rising edge of an index input.
Latch on Falling Edge (QEPCTL[IEL] = 10)	The position-counter value (QPOSCNT) is latched to the QPOSILAT register on every falling edge of index input.
Latch on Index Event Marker/Software Index Marker (QEPCTL[IEL] = 11	The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and the direction on the first index event marker (QEPSTS[FIDF]) in the QEPSTS registers. The eQEP peripheral also remembers the quadrature edge on the first index marker so that the same relative quadrature transition is used for latching the position counter (QEPCTL[IEL] = 11).

Figure 17-11 shows the position counter latch using an index event marker.

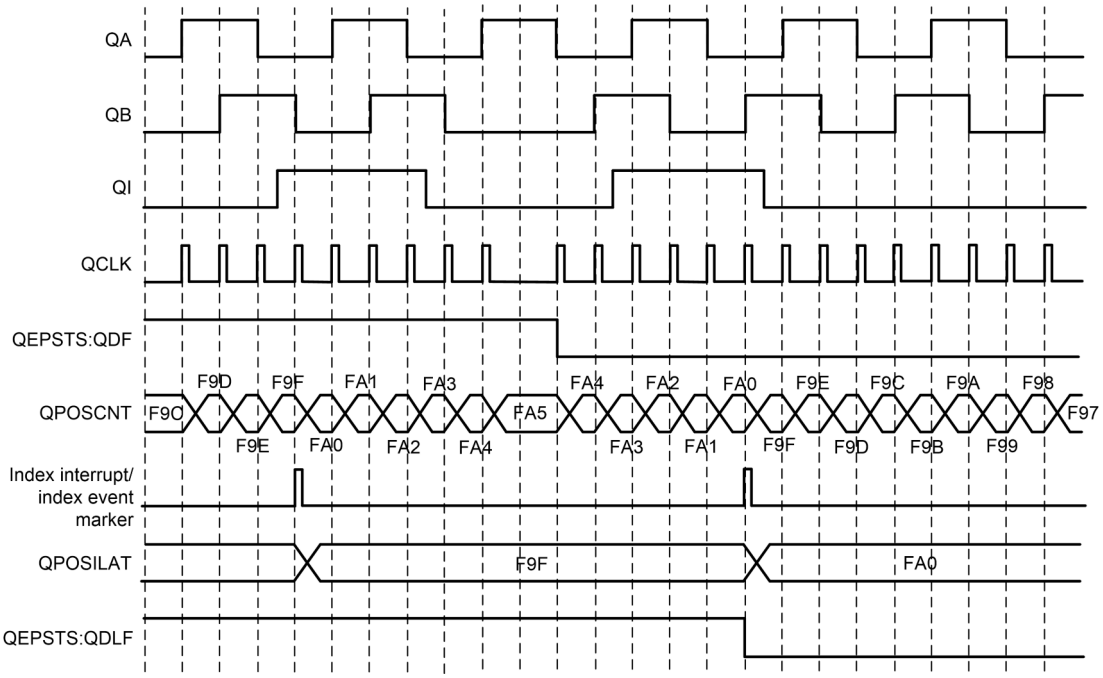


Figure 17-11. Software Index Marker for 1000-line Encoder (QEPCTL[IEL] = 1)

17.5.2.2 Strobe Event Latch

The position-counter value is latched to the QPOSSLAT register on the rising edge of the strobe input by clearing the QEPCTL[SEL] bit.

If the QEPCTL[SEL] bit is set, then the position-counter value is latched to the QPOSSLAT register on the rising edge of the strobe input for forward direction, and on the falling edge of the strobe input for reverse direction as shown in Figure 17-12.

The strobe event latch interrupt flag (QFLG[SEL) is set when the position counter is latched to the QPOSSLAT register.

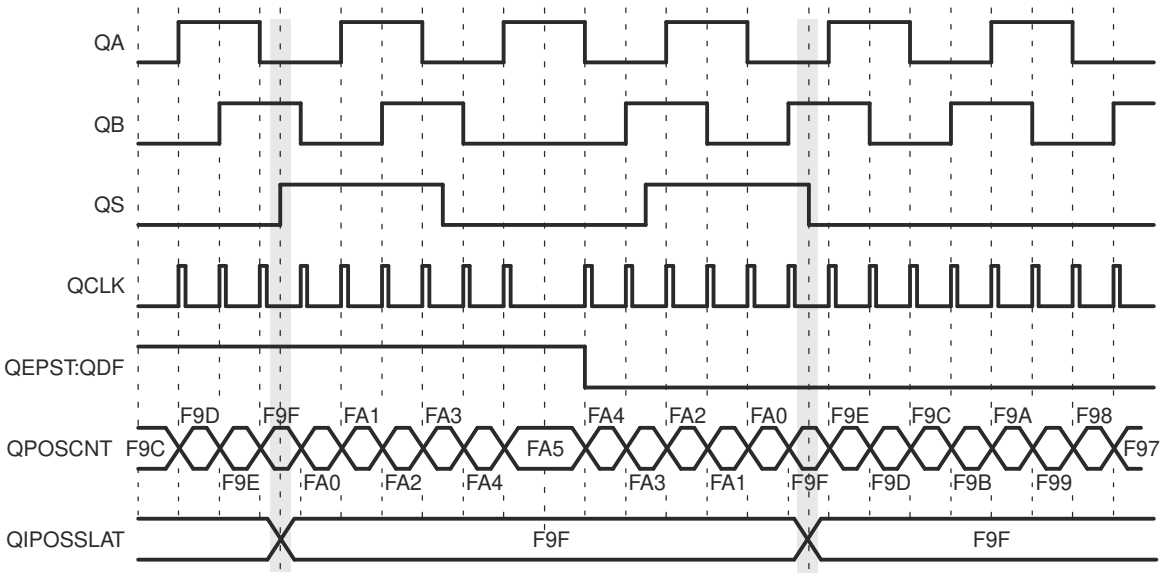


Figure 17-12. Strobe Event Latch (QEPCTL[SEL] = 1)

There is an added feature on Type 2.0 eQEP where position-counter value can also be latched on ADCSOCA and ADCSOSCB events by configuring the register QEPSTROBESEL.STROBESEL as shown in Figure 17-13. To use the ADCSOCA/B events for the QS signal, configuration of the QEPSRCSEL.QEPSSEL to be non-zero is needed..

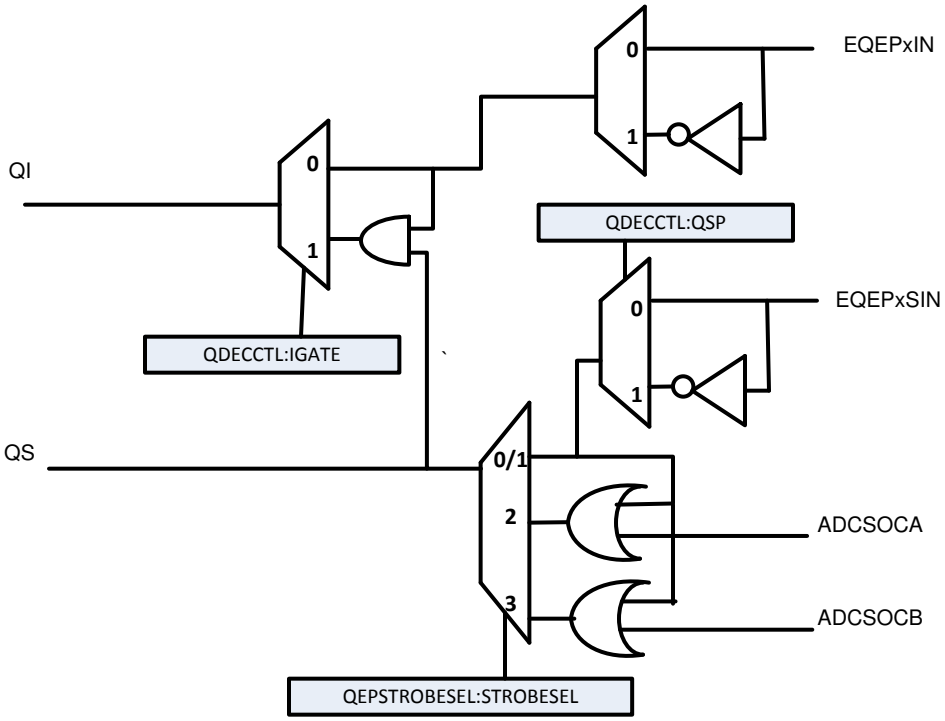


Figure 17-13. Latching Position Counter on ADCSOCA/ADCSOCB Event

17.5.3 Position Counter Initialization

The position counter can be initialized using the following events:

- Index event
- Strobe event
- Software initialization

Index Event Initialization (IEI)	The QEPI index input can be used to trigger the initialization of the position counter at the rising or falling edge of the index input. If the QEPCTL[IEI] bits are 10, then the position counter (QPOSCNT) is initialized with a value in the QPOSINIT register on the rising edge of index input. Conversely, if the QEPCTL[IEI] bits are 11, initialization is on the falling edge of the index input.
Strobe Event Initialization (SEI)	If the QEPCTL[SEI] bits are 10, then the position counter is initialized with a value in the QPOSINIT register on the rising edge of strobe input. If QEPCTL[SEL] bits are 11, then the position counter is initialized with a value in the QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.
Software Initialization (SWI)	The position counter can be initialized in software by writing a 1 to the QEPCTL[SWI] bit. This bit is not automatically cleared. While the bit is still set, if a 1 is written to the bit again, the position counter is re-initialized.

17.5.4 eQEP Position-compare Unit

The eQEP peripheral includes a position-compare unit that is used to generate a sync output and interrupt on a position-compare match. Figure 17-14 shows a diagram. The position-compare (QPOSCMP) register is shadowed and shadow mode can be enabled or disabled using the QPOSCTL[PSSHDW] bit. If the shadow mode is not enabled, the CPU writes directly to the active position compare register.

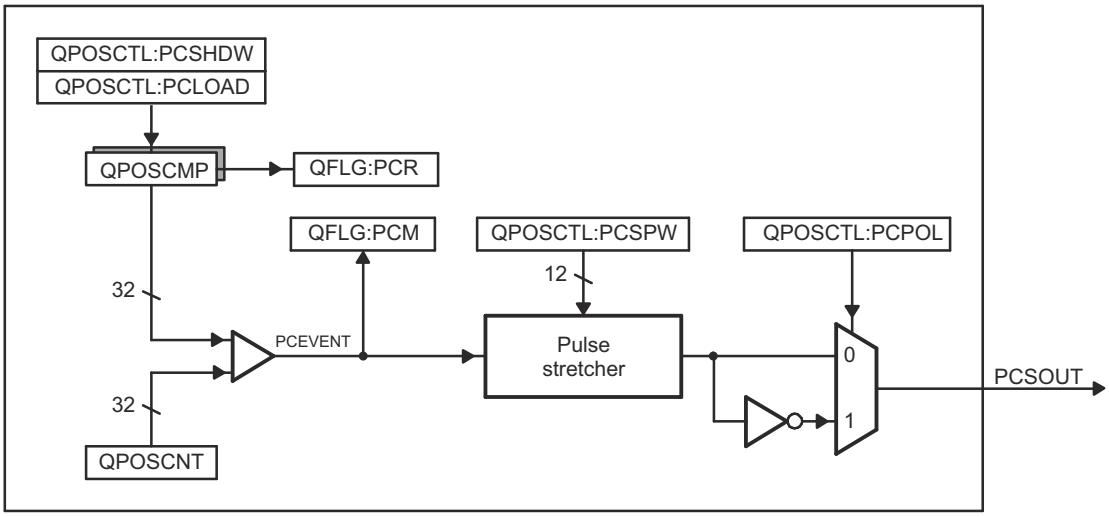


Figure 17-14. eQEP Position-compare Unit

In shadow mode, you can configure the position-compare unit (QPOSCTL[PCLOAD]) to load the shadow register value into the active register on the following events, and to generate the position-compare ready (QFLG[PCR]) interrupt after loading.

- Load on compare match
- Load on position-counter zero event

The position-compare match (QFLG[PCM]) is set when the position-counter value (QPOSCNT) matches with the active position-compare register (QPOSCMP) and the position-compare sync output of the programmable pulse width is generated on compare-match to trigger an external device.

For example, if QPOSCMP = 2, the position-compare unit generates a position-compare event on 1 to 2 transitions of the eQEP position counter for forward counting direction and on 3 to 2 transitions of the eQEP position counter for reverse counting direction (see Figure 17-15).

See the register section for the layout of the eQEP Position-Compare Control Register (QPOSCTL) and description of the QPOSCTL bit fields.

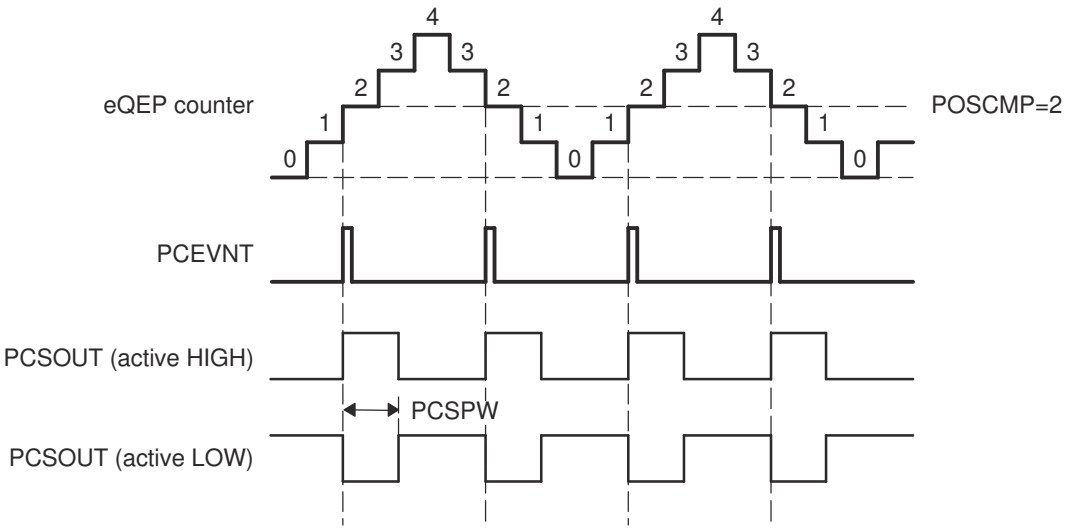


Figure 17-15. eQEP Position-compare Event Generation Points

The pulse stretcher logic in the position-compare unit generates a programmable position-compare sync pulse output on the position-compare match. In the event of a new position-compare match while a previous position-compare pulse is still active, then the pulse stretcher generates a pulse of specified duration from the new position-compare event as shown in Figure 17-16.

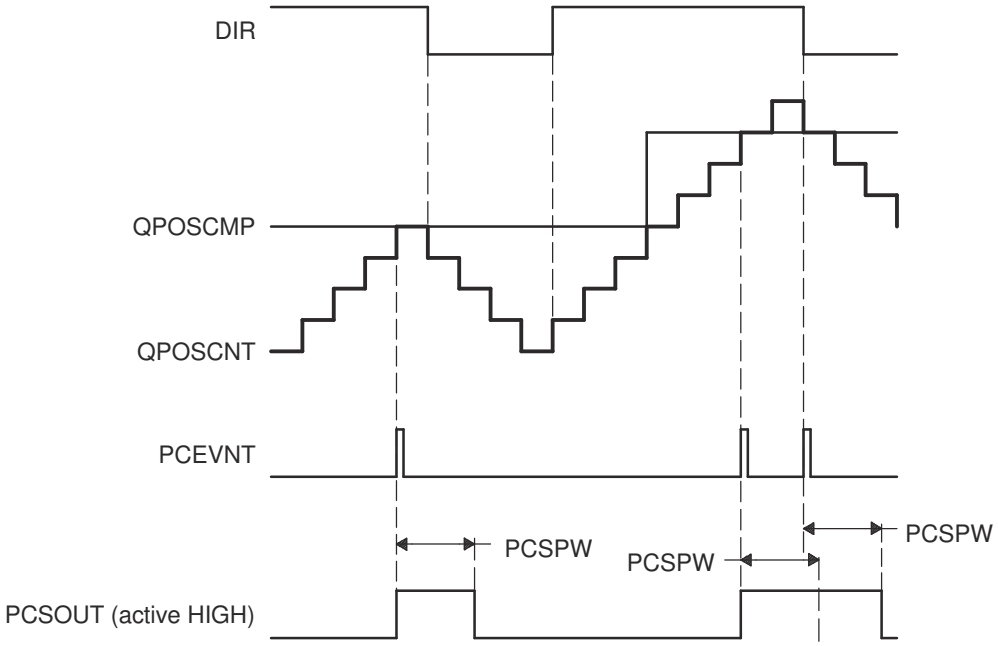


Figure 17-16. eQEP Position-compare Sync Output Pulse Stretcher

17.6 eQEP Edge Capture Unit

The eQEP peripheral includes an integrated edge capture unit to measure the elapsed time between the unit position events as shown in Figure 17-17. This feature is typically used for low-speed measurement using the following formula:

$$v(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (23)$$

where:

- X = Unit position is defined by integer multiple of quadrature edges (see Figure 17-18)
- ΔT = Elapsed time between unit position events
- v(k) = Velocity at time instant "k"

The eQEP capture timer (QCTMR) runs from prescaled SYSCLKOUT and the prescaler is programmed by the QCAPCTL[CCPS] bits. The capture timer (QCTMR) value is latched into the capture period register (QCPRD) on every unit position event and then the capture timer is reset, a flag is set in QEPSTS:UPEVNT to indicate that new value is latched into the QCPRD register. Software can check this status flag before reading the period register for low speed measurement, and clear the flag by writing 1.

Time measurement (ΔT) between unit position events is correct if the following conditions are met:

- No more than 65,535 counts have occurred between unit position events.
- No direction change between unit position events.

If the QEP capture timer overflows between unit position events, then the timer sets the QEP capture overflow flag (QEPSTS[COEF]) in the status register.

Note

The QCPRDLAT register is not set to 0xFFFF if the QEPSTS[COEF] bit gets set.

If direction change occurs between the unit position events, then the direction error flag (QEPSTS[CDEF]) is set in the status register and the QCPRDLAT register is set to 0xFFFF.

The Capture Timer (QCTMR) and Capture Period register (QCPRD) can be configured to latch on the following events:

- CPU read of QPOSCNT register
- Unit time-out event

If the QEPCTL[QCLM] bit is cleared, then the capture timer and capture period values are latched into the QCTMRLAT and QCPRDLAT registers, respectively, when the CPU reads the position counter (QPOSCNT).

If the QEPCTL[QCLM] bit is set, then the position counter, capture timer, and capture period values are latched into the QPOSLAT, QCTMRLAT and QCPRDLAT registers, respectively, on unit time out.

Figure 17-19 shows the capture unit operation along with the position counter.

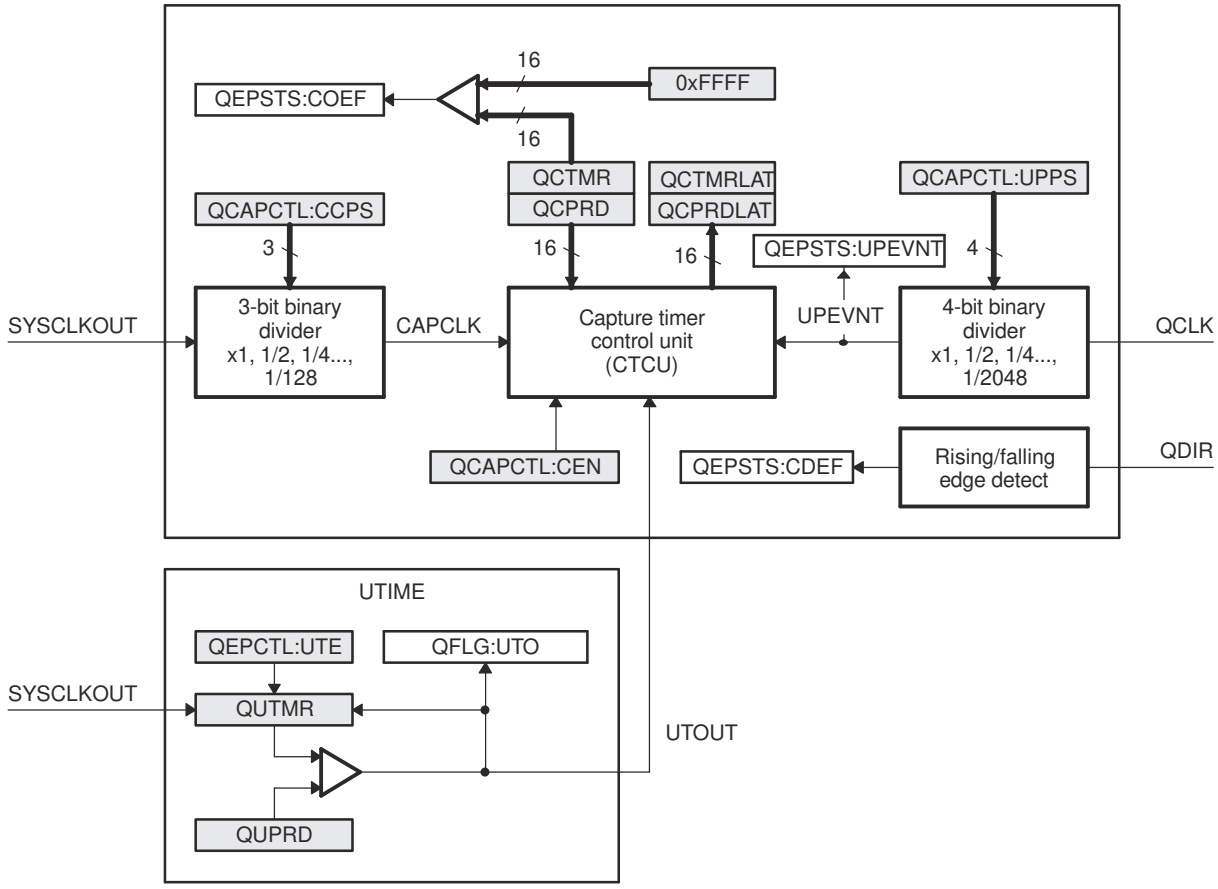
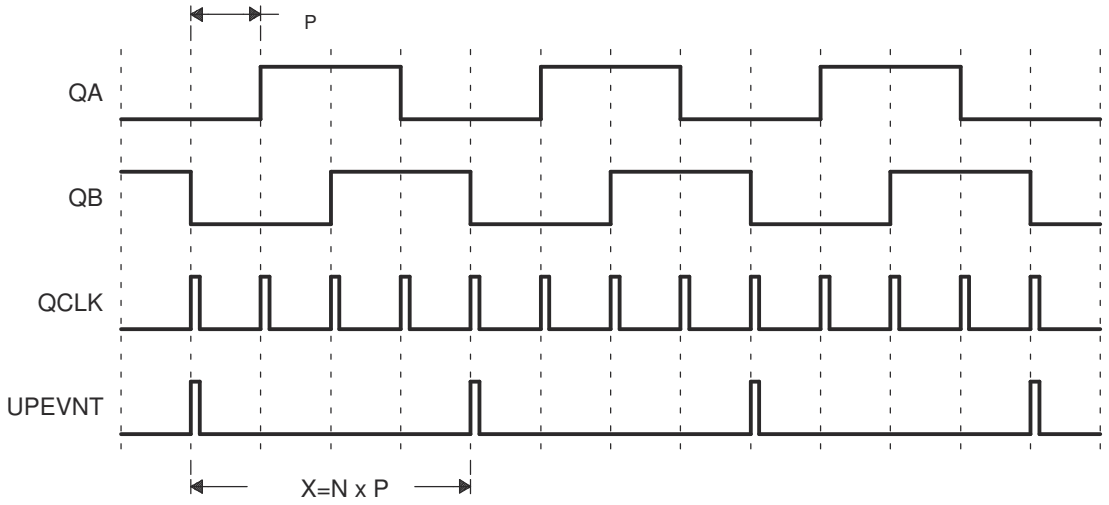


Figure 17-17. eQEP Edge Capture Unit

CAUTION

The QCAPCTL[UPPS] prescaler cannot be modified dynamically (such as switching the unit event prescaler from QCLK/4 to QCLK/8). Doing so can result in undefined behavior. The QCAPCTL[CPPS] prescaler can be modified dynamically (such as switching CAPCLK prescaling mode from SYSCLK/4 to SYSCLK/8) only after the capture unit is disabled.



N = Number of quadrature periods selected using QCAPCTL[UPPS] bits

Figure 17-18. Unit Position Event for Low Speed Measurement (QCAPCTL[UPPS] = 0010)

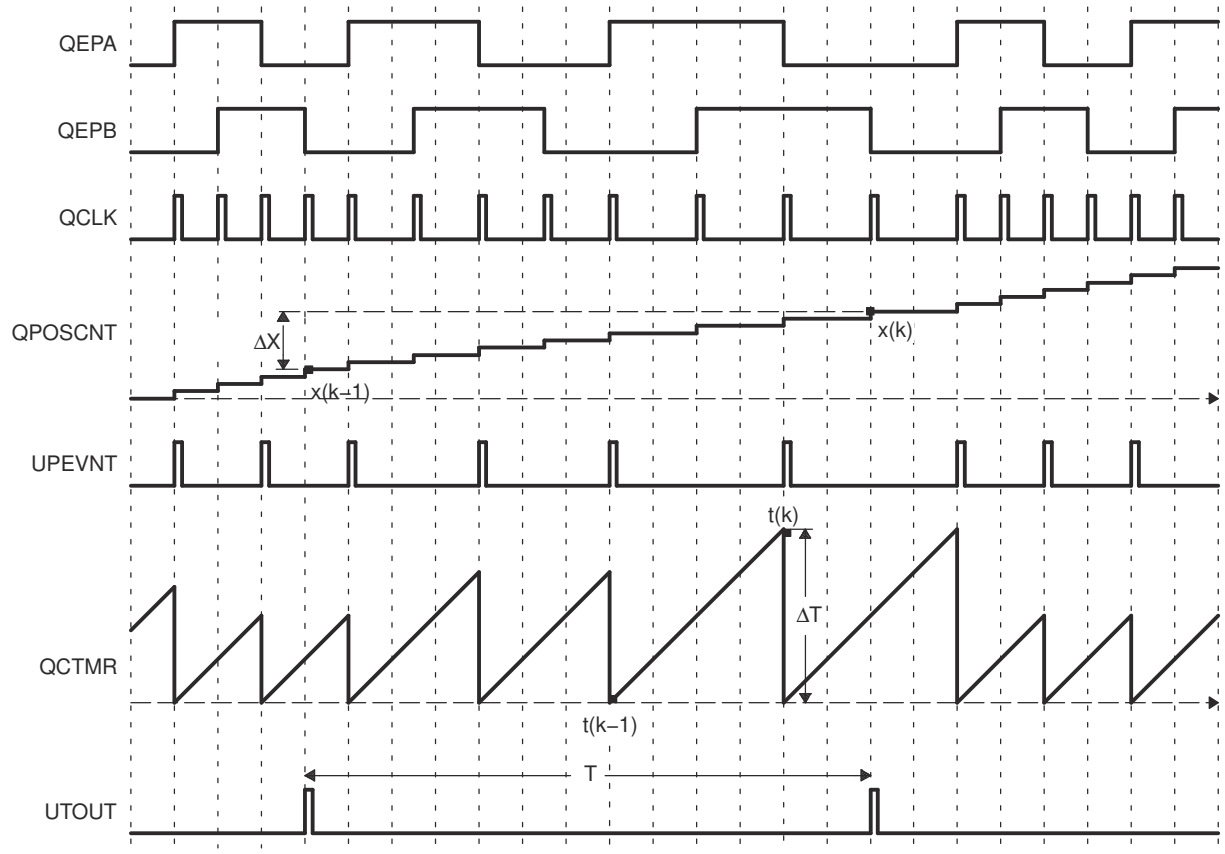


Figure 17-19. eQEP Edge Capture Unit - Timing Details

Velocity calculation equation:

$$v(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \text{ o} \quad (24)$$

where:

- v(k) = Velocity at time instant k
- x(k) = Position at time instant k
- x(k-1) = Position at time instant k-1
- T = Fixed unit time or inverse of velocity calculation rate
- ΔX = Incremental position movement in unit time
- X = Fixed unit position
- ΔT = Incremental time elapsed for unit position movement
- t(k) = Time instant "k"
- t(k-1) = Time instant "k-1"

Unit time (T) and unit period (X) are configured using the QUPRD and QCAPCTL[UPPS] registers. Incremental position output and incremental time output is available in the QPOSLAT and QCPRDLAT registers.

Parameter	Relevant Register to Configure or Read the Information
T	Unit Period Register (QUPRD)
ΔX	Incremental Position = QPOSLAT(k) - QPOSLAT(K-1)
X	Fixed-unit position defined by sensor resolution and QCAPCTL[UPPS] bits
ΔT	Capture Period Latch (QCPRDLAT)

17.7 eQEP Watchdog

The eQEP peripheral contains a 16-bit watchdog timer (Figure 17-20) that monitors the quadrature clock to indicate proper operation of the motion-control system. The eQEP watchdog timer is clocked from SYSCLKOUT/64 and the quadrature clock event (pulse) resets the watchdog timer. If no quadrature clock event is detected until a period match (QWDPRD = QWDTMR), then the watchdog timer times out and the watchdog interrupt flag is set (QFLG[WTO]). The time-out value is programmable through the watchdog period register (QWDPRD).

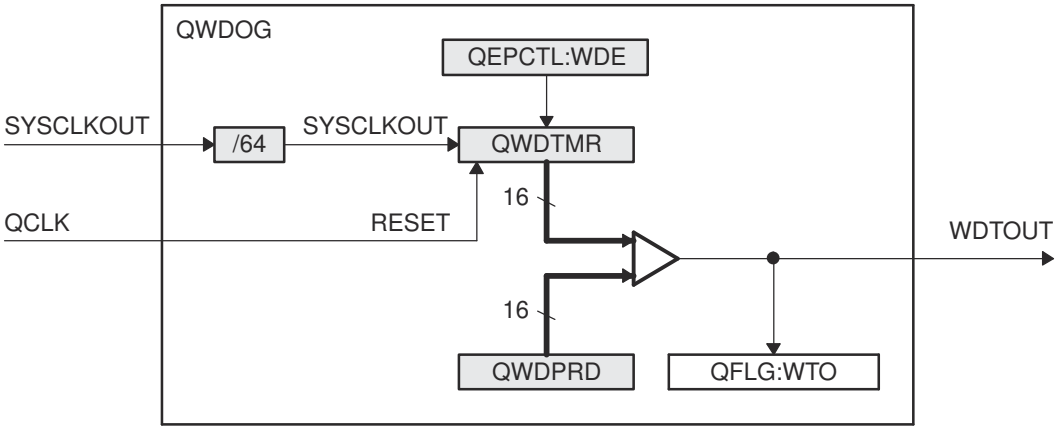


Figure 17-20. eQEP Watchdog Timer

17.8 eQEP Unit Timer Base

The eQEP peripheral includes a 32-bit timer (QUTMR) that is clocked by SYSCLKOUT to generate periodic interrupts for velocity calculations, see Figure 17-21. Whenever the unit timer (QUTMR) matches the unit period register (QUPRD), the eQEP peripheral resets the unit timer (QUTMR) and also generates the unit time out interrupt flag (QFLG[UTO]). The unit timer gets reset whenever timer value equals to configured period value.

The eQEP peripheral can be configured to latch the position counter, capture timer, and capture period values on a unit time out event so that latched values are used for velocity calculation as described in Section 17.6.

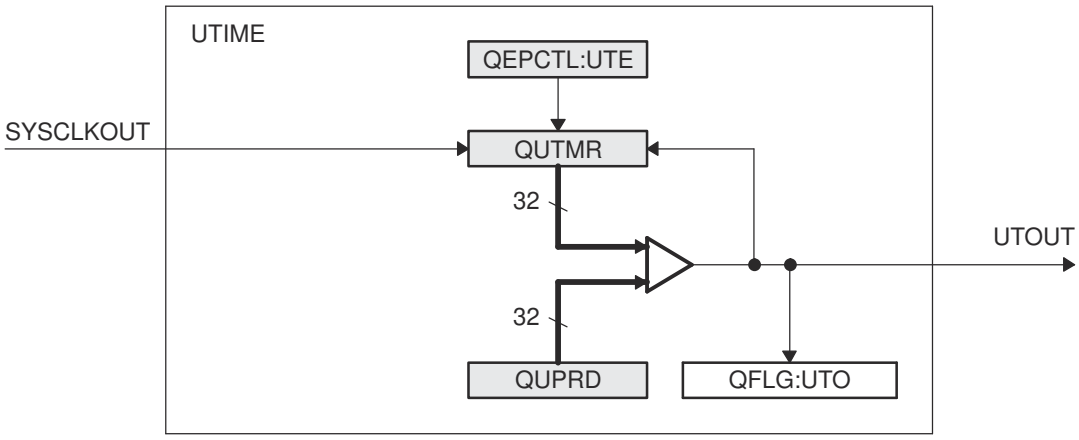


Figure 17-21. eQEP Unit Timer Base

17.9 QMA Module

The QEP Mode Adapter (QMA) is designed to extend the C2000™ eQEP module capabilities to support the additional modes described. Figure 17-22 depicts how the QMA module is integrated into to the eQEP module.

At reset, by default QMA logic is bypassed and the EQEPA and EQEPB inputs from the pins go directly into the eQEP module. When QMA module is enabled by configuring the QMACTRL[MODE] register, the EQEPA and EQEPB input are processed by this module and modified version of EQEPA and EQEPB signals are sent to the eQEP module. The QMA module requires the eQEP module to be configured in the Direction-Count mode and generates a clock signal on EQEPA input and direction signal on EQEPB input as needed for the proper operation of the intended mode.

- The xCLKMOD block inside the QMA module looks at the transitions on external EQEPA and EQEPB signals to generate the clock signal on the EQEPA input to the eQEP module.
- The xDIRMOD block inside the QMA module looks at the transitions on external EQEPA and EQEPB signals to generate the direction signal on the EQEPB input to the eQEP module.

The QMA module has error detection logic to detect illegal transitions on EQEPA and EQEPB input signals. The QMA module's error and interrupt are integrated inside the eQEP module as described in Section 17.10. In addition, the QMACTRL register configuration can be locked using the QMALOCK register. Refer to the register description for more details.

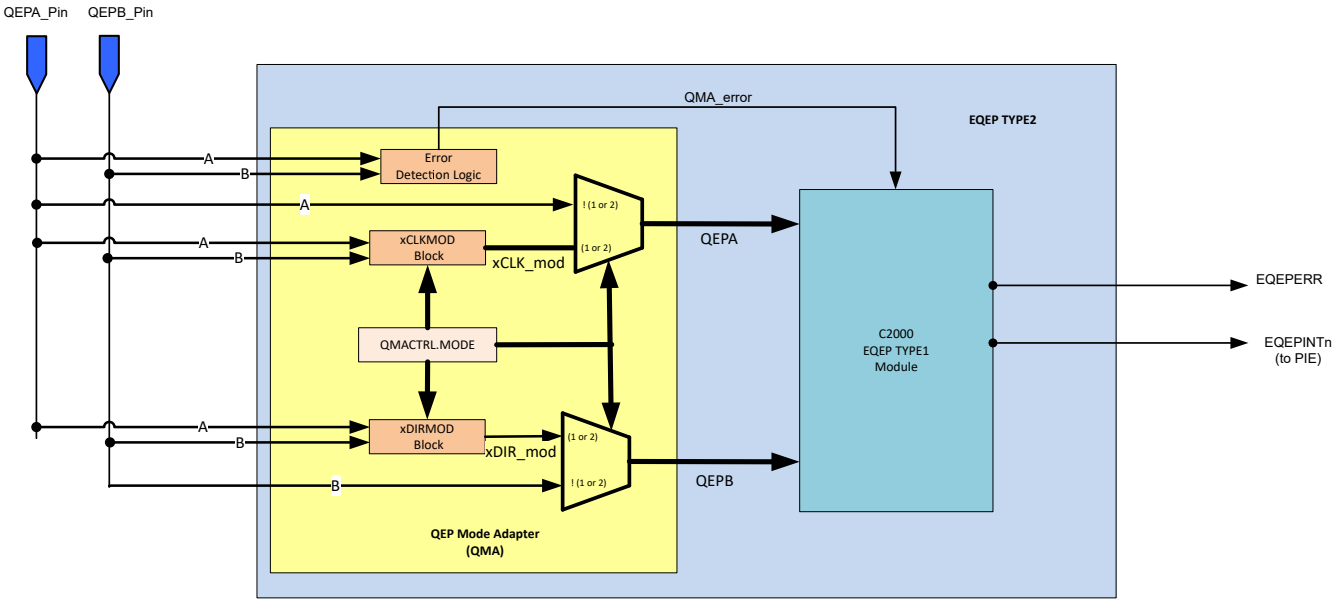


Figure 17-22. QMA Module Block Diagram

17.9.1 Modes of Operation

The QMA module can be operated in the following modes by configuring the QMACTRL register:

- QMA Mode-1 (QMACTRL[MODE] = 1)
- QMA Mode-2 (QMACTRL[MODE] = 2)

17.9.1.1 QMA Mode-1 (QMACTRL[MODE] = 1)

This mode is used when the default state of EQEPA and EQEPB inputs is high. In this mode, outputs of QMA correspond to the following as shown in Figure 17-23:

- EQEPA Output of QMA is the AND of EQEPA and EQEPB inputs coming from the pin
- EQEPB Output of QMA is the direction signal generated by QMA based on EQEPA and EQEPB inputs

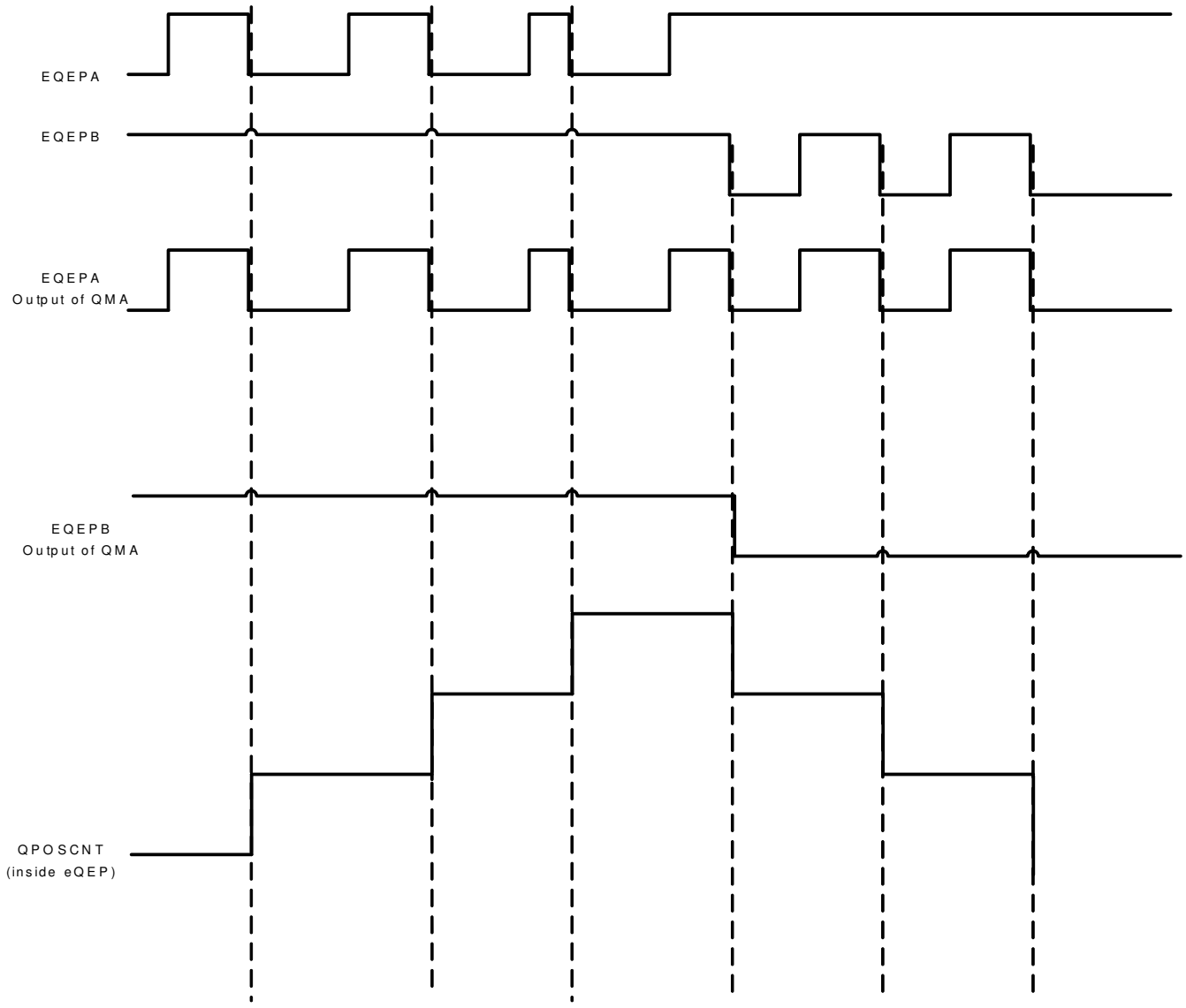


Figure 17-23. QMA Mode-1

17.9.1.2 QMA Mode-2 (QMACTRL[MODE] = 2)

This mode is used when the default state of EQEPA and EQEPB inputs is low. In this mode, outputs of QMA correspond to the following as shown in Figure 17-24:

- EQEPA Output of QMA is the OR of EQEPA and EQEPB inputs coming from the pin
- EQEPB Output of QMA is the direction signal generated by QMA based on EQEPA and EQEPB inputs

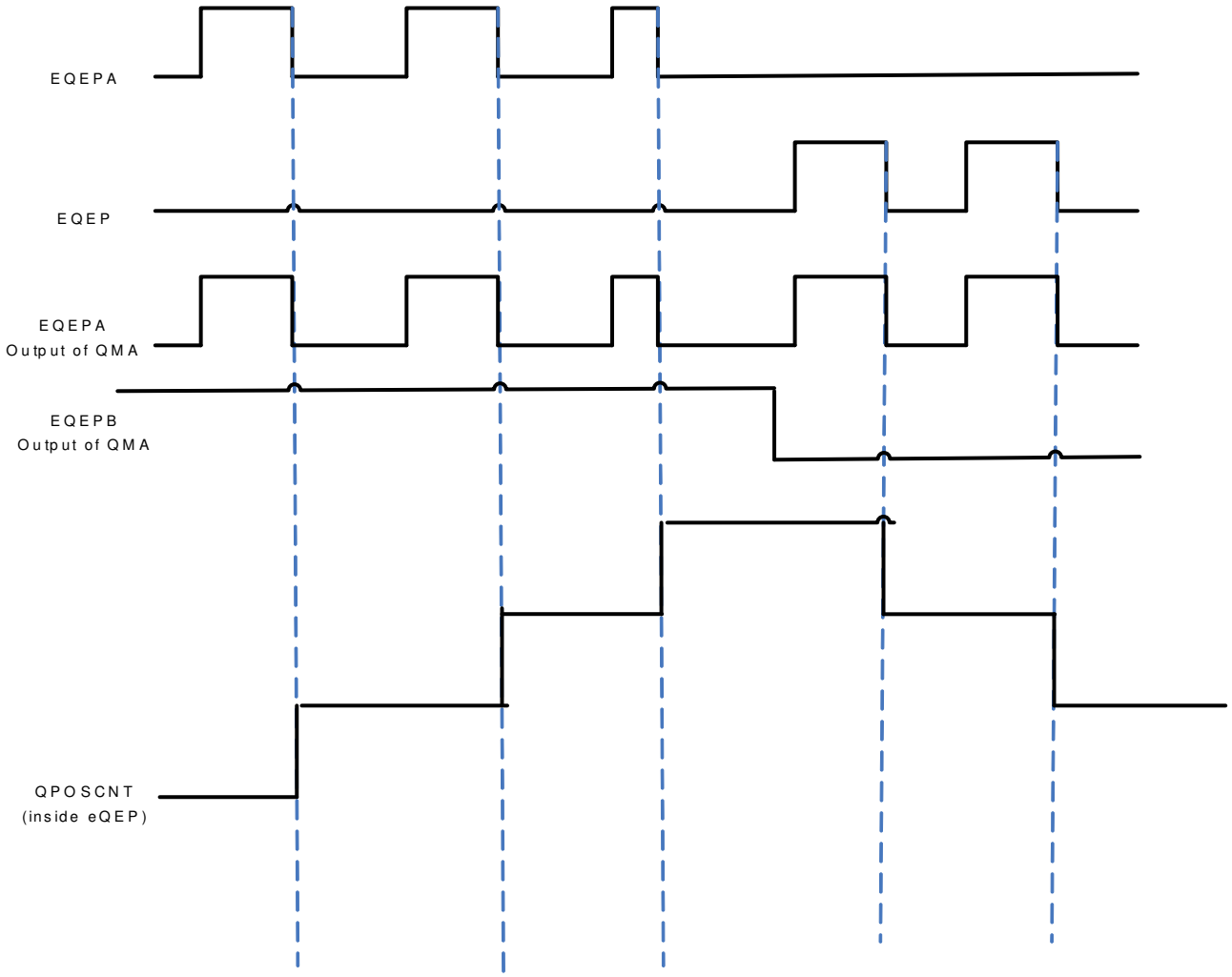


Figure 17-24. QMA Mode-2

17.9.2 Interrupt and Error Generation

The error detection logic detects illegal transitions on EQEPA and EQEPB signals and generates an error signal. This error signal can be used to generate eQEP interrupt and error output. Refer to Section 17.10 for details.

17.10 eQEP Interrupt Structure

Figure 17-25 shows how the interrupt mechanism works in the eQEP module.

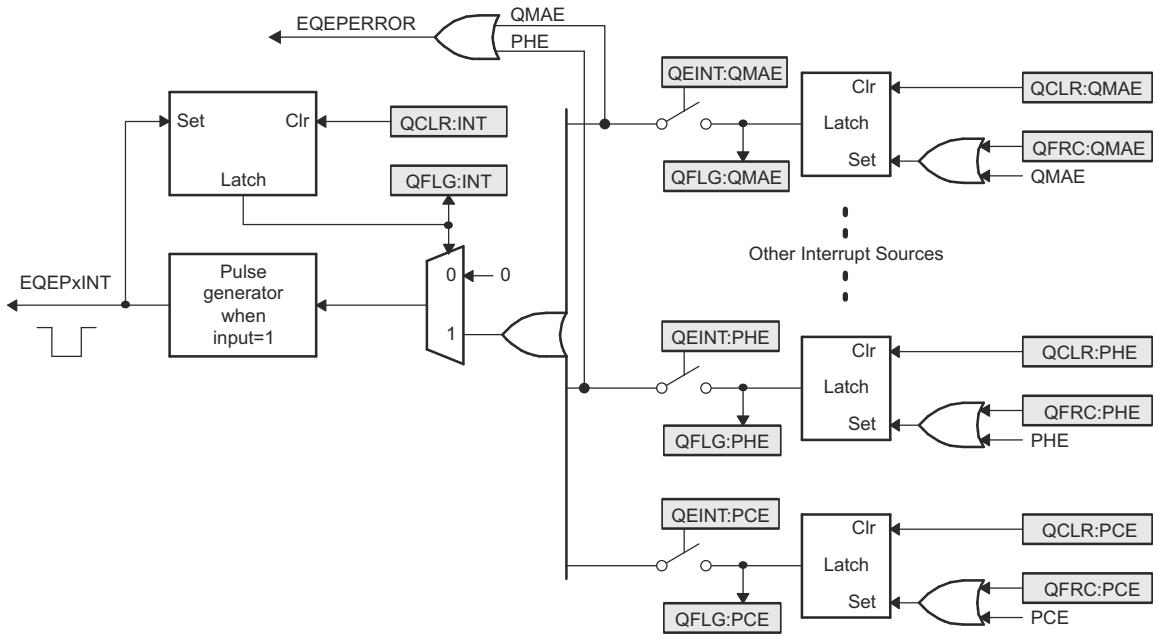


Figure 17-25. eQEP Interrupt Generation

Eleven interrupt events (PCE, PHE, QDC, WTO, PCU, PCO, PCR, PCM, SEL, IEL, and UTO) can be generated. The interrupt control register (QEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (QFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT).

A pulse interrupt is generated to the PIE when the following conditions are met:

1. Interrupt is enabled for eQEP event inside QEINT register
2. Interrupt flag for eQEP event inside QFLG register is set, and
3. Global interrupt status flag bit QFLG[INT] had been cleared for previously generated interrupt event. The interrupt service routine needs to clear the global interrupt flag bit and the serviced event, by way of the interrupt clear register (QCLR), before any other interrupt pulses are generated. If either flags inside the QFLG register are not cleared, further interrupt events do not generate an additional interrupt."
4. You can force an interrupt event by way of the interrupt force register (QFRC), which is useful for test/debug purposes.

17.11 Software

17.11.1 EQEP Examples

NOTE: These examples are located in the C2000Ware installation at the following location: C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/eqep

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

17.11.1.1 Frequency Measurement Using eQEP

FILE: eqep_ex1_freq_cal.c

This example calculates the frequency of an input signal using the eQEP module. MCPWM1A is configured to generate this input signal with a frequency of 5kHz. There is an interrupt once every period with a call to the frequency calculation function. This example uses the IQMath library to simplify high-precision calculations.

In addition to the main example file, the following files must be included in this project:
- *eqep_ex1_calculation.c* - contains frequency calculation function
- *eqep_ex1_calculation.h* - includes initialization values for frequency structure

The configuration for this example is as follows:
- Maximum frequency is configured to 10KHz (baseFreq)
- Minimum frequency is assumed at 50Hz for capture pre-scalar selection

SPEED_FR: High Frequency Measurement is obtained by counting the external input pulses for 10ms (unit timer set to 100Hz). \[SPEED_FR = \frac{Count\ Delta}{10ms} \]

SPEED_PR: Low Frequency Measurement is obtained by measuring time period of input edges. Time measurement is averaged over 64 edges for better results and the capture unit performs the time measurement using pre-scaled SYSCLK.

Note that the pre-scaler for capture unit clock is selected such that the capture timer does not overflow at the required minimum frequency. This example runs indefinitely until the user stops the timer.

For more information about the frequency calculation see the comments at the beginning of eqep_ex1_calculation.c and the XLS file provided with the project, eqep_ex1_calculation.xls.

External Connections
- Connect GPIO6/eQEP1A to GPIO0/MCPWM1A

Watch Variables
- *freq.freqHzFR* - Frequency measurement using position counter/unit time out
- *freq.freqHzPR* - Frequency measurement using capture unit

17.11.1.2 Position and Speed Measurement Using eQEP

FILE: eqep_ex2_pos_speed.c

This example provides position and speed measurement using the capture unit and speed measurement using unit time out of the eQEP module. MCPWM1 and a GPIO are configured to generate simulated eQEP signals. The MCPWM module interrupts once every period and calls the position/speed calculation function. This example uses the IQMath library to simplify high-precision calculations.

In addition to the main example file, the following files must be included in this project:
- *eqep_ex2_calculation.c* - contains position/speed calculation function
- *eqep_ex2_calculation.h* - includes initialization values for position/speed structure

The configuration for this example is as follows:
- Maximum speed is configured to 6000rpm (baseRPM)
- Minimum speed is assumed at 10rpm for capture pre-scalar selection
- Pole pair is configured to 2 (polePairs)

- Encoder resolution is configured to 4000 counts/revolution (mechScaler)
- Which means: 4000 / 4 = 1000 line/revolution quadrature encoder (simulated by MCPWM1)
- MCPWM1 (simulating QEP encoder signals) is configured for a 5kHz frequency or 300 rpm (= 4 * 5000 cnts/sec * 60 sec/min) / 4000 cnts/rev)

SPEEDRPM_FR: High Speed Measurement is obtained by counting the QEP input pulses for 10ms (unit timer set to 100Hz).

SPEEDRPM_FR = (Position Delta / 10ms) * 60 rpm

SPEEDRPM_PR: Low Speed Measurement is obtained by measuring time period of QEP edges. Time measurement is averaged over 64 edges for better results and the capture unit performs the time measurement using pre-scaled SYSCLK .

Note that the pre-scaler for capture unit clock is selected such that the capture timer does not overflow at the required minimum frequency. This example runs indefinitely until the user ends the program.

For more information about the position/speed calculation see the comments at the beginning of eqep_ex2_calculation.c and the XLS file provided with the project, eqep_ex2_calculation.xls.

External Connections
- Connect GPIO6/eQEP1A to GPIO0/MCPWM1A (simulates eQEP Phase A signal)
- Connect GPIO7/eQEP1B to GPIO1/MCPWM1B (simulates eQEP Phase B signal)
- Connect GPIO43/eQEP1I to GPIO2 (simulates eQEP Index Signal)

Watch Variables
- *posSpeed.speedRPMFR* - Speed meas. in rpm using QEP position counter
- *posSpeed.speedRPMPR* - Speed meas. in rpm using capture unit
- *posSpeed.thetaMech* - Motor mechanical angle (Q15)
- *posSpeed.thetaElec* - Motor electrical angle (Q15)

17.11.1.3 Frequency Measurement Using eQEP via unit timeout interrupt

FILE: eqep_ex4_freq_cal_interrupt.c

This example will calculate the frequency of an input signal using the eQEP module. MCPWM1A is configured to generate this input signal with a frequency of 5kHz. EQEP unit timeout is set which generates an interrupt every *UNIT_PERIOD* microseconds and frequency calculation occurs continuously.

The configuration for this example is as follows:
- PWM frequency is specified as 5000Hz
- UNIT_PERIOD is specified as 10000µs
- Min frequency is (1 / (2*10ms)), that is, 50Hz
- Highest frequency can be (2^32) / ((2*10ms))
- Resolution of frequency measurement is 50Hz

freq : Frequency Measurement is obtained by counting the external input pulses for UNIT_PERIOD (unit timer set to 10ms).

External Connections for Control Card
- Connect GPIO6/eQEP1A to GPIO0/MCPWM1A

Watch Variables
- *freq* - Frequency measurement using position counter/unit time out
- *pass* - If measured frequency matches with PWM frequency then pass = 1 else 0

17.11.1.4 Motor speed and direction measurement using eQEP via unit timeout interrupt

FILE: eqep_ex5_speed_dir_motor.c

This example can be used to sense the speed and direction of motor using eQEP in quadrature encoder mode. MCPWM1A is configured to simulate motor encoder signals with frequency of 5kHz on both A and B pins with

90 degree phase shift (so as to run this example without motor). EQEP unit timeout is set which will generate an interrupt every *UNIT_PERIOD* microseconds and speed calculation occurs continuously based on the direction of motor

The configuration for this example is as follows

- PWM frequency is specified as 5000Hz
- UNIT_PERIOD is specified as 10000µs
- Simulated quadrature signal frequency is 20000Hz (4 * 5000)
- Encoder holes assumed as 1000
- Thus Simulated motor speed is 300rpm (5000 * (60 / 1000))

freq : Simulated quadrature signal frequency measured by counting the external input pulses for UNIT_PERIOD (unit timer set to 10ms). *speed* : Measure motor speed in rpm *dir* : Indicates clockwise (1) or anticlockwise (-1)

External Connections (if motor encoder signals are simulated by MCPWM)
With motor

- Comment in "MOTOR" in includes
- Connect GPIO6/eQEP1A to GPIO0/MCPWM1A (simulates eQEP Phase A signal)
- Connect GPIO7/eQEP1B to GPIO1/MCPWM1B (simulates eQEP Phase B signal)

Watch Variables

- *freq* : Simulated motor frequency measurement is obtained by counting the external input pulses for UNIT_PERIOD (unit timer set to 10ms).
- *speed* : Measure motor speed in rpm
- *dir* : Indicates clockwise (1) or anticlockwise (-1)
- *pass* - If measured quadrature frequency matches with input quadrature frequency (4 * PWM frequency) then pass = 1 else fail = 1 (** only when "MOTOR" is commented out)

17.12 EQEP Registers

This Section describes the EQEP Registers.

17.12.1 EQEP Base Address Table

Table 17-4. EQEP Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
EQep1Regs	EQEP_REGS	EQEP1_BASE	0x0000_5100	YES	YES

17.12.2 EQEP_REGS Registers

Table 17-5 lists the memory-mapped registers for the EQEP_REGS registers. All register offset addresses not listed in Table 17-5 should be considered as reserved locations and the register contents should not be modified.

Table 17-5. EQEP_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	QPOSCNT	Position Counter	
2h	QPOSINIT	Position Counter Init	
4h	QPOSMAX	Maximum Position Count	
6h	QPOSCMP	Position Compare	
8h	QPOSILAT	Index Position Latch	
Ah	QPOSSLAT	Strobe Position Latch	
Ch	QPOSLAT	Position Latch	
Eh	QUTMR	QEP Unit Timer	
10h	QUPRD	QEP Unit Period	
12h	QWDTMR	QEP Watchdog Timer	
13h	QWDPRD	QEP Watchdog Period	
14h	QDECCTL	Quadrature Decoder Control	
15h	QEPCTL	QEP Control	
16h	QCAPCTL	Qaudrature Capture Control	
17h	QPOSCTL	Position Compare Control	
18h	QEINT	QEP Interrupt Control	
19h	QFLG	QEP Interrupt Flag	
1Ah	QCLR	QEP Interrupt Clear	
1Bh	QFRC	QEP Interrupt Force	
1Ch	QEPSTS	QEP Status	
1Dh	QCTMR	QEP Capture Timer	
1Eh	QCPRD	QEP Capture Period	
1Fh	QCTMRLAT	QEP Capture Latch	
20h	QCPRDLAT	QEP Capture Period Latch	
30h	REV	QEP Revision Number	
32h	QEPSTROBESEL	QEP Strobe select register	
34h	QMACTRL	QMA Control register	
36h	QEPSRCSEL	QEP Source Select Register	

Complex bit access types are encoded to fit into small table cells. Table 17-6 shows the codes that are used for access types in this section.

Table 17-6. EQEP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear

Table 17-6. EQEP_REGS Access Type Codes (continued)

Access Type	Code	Description
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

17.12.2.1 QPOSCNT Register (Offset = 0h) [Reset = 00000000h]

QPOSCNT is shown in Figure 17-26 and described in Table 17-7.

Return to the Summary Table.

Position Counter

Figure 17-26. QPOSCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCNT																															
R/W-0h																															

Table 17-7. QPOSCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSCNT	R/W	0h	Position Counter This 32-bit position counter register counts up/down on every eQEP pulse based on direction input. This counter acts as a position integrator whose count value is proportional to position from a give reference point. This Register acts as a Read ONLY register while counter is counting up/down. Note: It is recommended to only write to the position counter register (QPOSCNT) during initialization, i.e. when the eQEP position counter is disabled (QPEN bit of QEPCTL is zero). Once the position counter is enabled (QPEN bit is one), writing to the eQEP position counter register (QPOSCNT) may cause unexpected results. Reset type: SYSRSn

17.12.2.2 QPOSINIT Register (Offset = 2h) [Reset = 00000000h]

QPOSINIT is shown in Figure 17-27 and described in Table 17-8.

Return to the Summary Table.

Position Counter Init

Figure 17-27. QPOSINIT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSINIT																															
R/W-0h																															

Table 17-8. QPOSINIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSINIT	R/W	0h	Position Counter Init This register contains the position value that is used to initialize the position counter based on external strobe or index event. The position counter can be initialized through software. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

17.12.2.3 QPOSMAX Register (Offset = 4h) [Reset = 00000000h]

QPOSMAX is shown in Figure 17-28 and described in Table 17-9.

Return to the Summary Table.

Maximum Position Count

Figure 17-28. QPOSMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSMAX																															
R/W-0h																															

Table 17-9. QPOSMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSMAX	R/W	0h	Maximum Position Count This register contains the maximum position counter value. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

17.12.2.4 QPOSCMP Register (Offset = 6h) [Reset = 00000000h]

QPOSCMP is shown in Figure 17-29 and described in Table 17-10.

Return to the Summary Table.

Position Compare

Figure 17-29. QPOSCMP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCMP																															
R/W-0h																															

Table 17-10. QPOSCMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSCMP	R/W	0h	Position Compare The position-compare value in this register is compared with the position counter (QPOSCNT) to generate sync output and/or interrupt on compare match. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

17.12.2.5 QPOSILAT Register (Offset = 8h) [Reset = 00000000h]

QPOSILAT is shown in Figure 17-30 and described in Table 17-11.

Return to the Summary Table.

Index Position Latch

Figure 17-30. QPOSILAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSILAT																															
R-0h																															

Table 17-11. QPOSILAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSILAT	R	0h	Index Position Latch The position-counter value is latched into this register on an index event as defined by the QEPCTL[IEL] bits. Reset type: SYSRSn

17.12.2.6 QPOSSLAT Register (Offset = Ah) [Reset = 00000000h]

QPOSSLAT is shown in Figure 17-31 and described in Table 17-12.

Return to the Summary Table.

Strobe Position Latch

Figure 17-31. QPOSSLAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSSLAT																															
R-0h																															

Table 17-12. QPOSSLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSSLAT	R	0h	Strobe Position Latch The position-counter value is latched into this register on a strobe event as defined by the QEPCTL[SEL] bits. Reset type: SYSRSn

17.12.2.7 QPOSLAT Register (Offset = Ch) [Reset = 00000000h]

QPOSLAT is shown in Figure 17-32 and described in Table 17-13.

Return to the Summary Table.

Position Latch

Figure 17-32. QPOSLAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSLAT																															
R-0h																															

Table 17-13. QPOSLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSLAT	R	0h	Position Latch The position-counter value is latched into this register on a unit time out event. Reset type: SYSRSn

17.12.2.8 QUTMR Register (Offset = Eh) [Reset = 00000000h]

QUTMR is shown in Figure 17-33 and described in Table 17-14.

Return to the Summary Table.

QEP Unit Timer

Figure 17-33. QUTMR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUTMR																															
R/W-0h																															

Table 17-14. QUTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QUTMR	R/W	0h	QEP Unit Timer This register acts as time base for unit time event generation. When this timer value matches the unit time period value a unit time event is generated. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

17.12.2.9 QUPRD Register (Offset = 10h) [Reset = 00000000h]

QUPRD is shown in Figure 17-34 and described in Table 17-15.

Return to the Summary Table.

QEP Unit Period

Figure 17-34. QUPRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUPRD																															
R/W-0h																															

Table 17-15. QUPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QUPRD	R/W	0h	QEP Unit Period This register contains the period count for the unit timer to generate periodic unit time events. These events latch the eQEP position information at periodic intervals and optionally generate an interrupt. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

17.12.2.10 QWDTMR Register (Offset = 12h) [Reset = 0000h]

QWDTMR is shown in Figure 17-35 and described in Table 17-16.

Return to the Summary Table.

QEP Watchdog Timer

Figure 17-35. QWDTMR Register

15	14	13	12	11	10	9	8
QWDTMR							
R/W-0h							
7	6	5	4	3	2	1	0
QWDTMR							
R/W-0h							

Table 17-16. QWDTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QWDTMR	R/W	0h	QEP Watchdog Timer This register acts as time base for the watchdog to detect motor stalls. When this timer value matches with the watchdog's period value a watchdog timeout interrupt is generated. This register is reset upon edge transition in quadrature-clock indicating the motion. Reset type: SYSRSn

17.12.2.11 QWDPRD Register (Offset = 13h) [Reset = 0000h]

QWDPRD is shown in Figure 17-36 and described in Table 17-17.

Return to the Summary Table.

QEP Watchdog Period

Figure 17-36. QWDPRD Register

15	14	13	12	11	10	9	8
QWDPRD							
R/W-0h							
7	6	5	4	3	2	1	0
QWDPRD							
R/W-0h							

Table 17-17. QWDPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QWDPRD	R/W	0h	QEP Watchdog Period This register contains the time-out count for the eQEP peripheral watch dog timer. When the watchdog timer value matches the watchdog period value, a watchdog timeout interrupt is generated. Reset type: SYSRSn

17.12.2.12 QDECCTL Register (Offset = 14h) [Reset = 0000h]

QDECCTL is shown in Figure 17-37 and described in Table 17-18.

Return to the Summary Table.

Quadrature Decoder Control

Figure 17-37. QDECCTL Register

15	14	13	12	11	10	9	8
QSRC		SOEN	SPSEL	XCR	SWAP	IGATE	QAP
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
QBP	QIP	QSP	RESERVED				QIDIRE
R/W-0h	R/W-0h	R/W-0h	R-0h				R/W-0h

Table 17-18. QDECCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	QSRC	R/W	0h	Position-counter source selection Reset type: SYSRSn 0h (R/W) = Quadrature count mode (QCLK = iCLK, QDIR = iDIR) 1h (R/W) = Direction-count mode (QCLK = xCLK, QDIR = xDIR) 2h (R/W) = UP count mode for frequency measurement (QCLK = xCLK, QDIR = 1) 3h (R/W) = DOWN count mode for frequency measurement (QCLK = xCLK, QDIR = 0)
13	SOEN	R/W	0h	Sync output-enable Reset type: SYSRSn 0h (R/W) = Disable position-compare sync output 1h (R/W) = Enable position-compare sync output
12	SPSEL	R/W	0h	Sync output pin selection Reset type: SYSRSn 0h (R/W) = Index pin is used for sync output 1h (R/W) = Strobe pin is used for sync output
11	XCR	R/W	0h	External Clock Rate Reset type: SYSRSn 0h (R/W) = 2x resolution: Count the rising/falling edge 1h (R/W) = 1x resolution: Count the rising edge only
10	SWAP	R/W	0h	CLK/DIR Signal Source for Position Counter Reset type: SYSRSn 0h (R/W) = Quadrature-clock inputs are not swapped 1h (R/W) = Quadrature-clock inputs are swapped
9	IGATE	R/W	0h	Index pulse gating option Reset type: SYSRSn 0h (R/W) = Disable gating of Index pulse 1h (R/W) = Gate the index pin with strobe
8	QAP	R/W	0h	QEPA input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPA input
7	QBP	R/W	0h	QEPB input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPB input
6	QIP	R/W	0h	QEPI input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPI input

Table 17-18. QDECCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	QSP	R/W	0h	QEPS input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPS input
4-1	RESERVED	R	0h	Reserved
0	QIDIRE	R/W	0h	0 - Compatible mode, Behavior same as existing devices 1 - Enhancement for Direction change during Index will be enabled: On QEPI direction change, the incoming posedge of QA can erroneously update/reset the position counter of the eQEP. This bit only needs to be enabled if the application requires a direction change occurring at the same time as an incoming QEPI signal, or when erroneous PC resets are observed. Reset type: SYSRSn

17.12.2.13 QEPCTL Register (Offset = 15h) [Reset = 0000h]

QEPCTL is shown in Figure 17-38 and described in Table 17-19.

Return to the Summary Table.

QEP Control

Figure 17-38. QEPCTL Register

15	14	13	12	11	10	9	8
FREE_SOFT		PCRM		SEI		IEI	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SWI	SEL	IEL		QPEN	QCLM	UTE	WDE
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 17-19. QEPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation mode Reset type: SYSRSn 0h (R/W) = QPOSCNT behavior Position counter stops immediately on emulation suspend 0h (R/W) = QWDTMR behavior Watchdog counter stops immediately 0h (R/W) = QUTMR behavior Unit timer stops immediately 0h (R/W) = QCTMR behavior Capture Timer stops immediately 1h (R/W) = QPOSCNT behavior Position counter continues to count until the rollover 1h (R/W) = QWDTMR behavior Watchdog counter counts until WD period match roll over 1h (R/W) = QUTMR behavior Unit timer counts until period rollover 1h (R/W) = QCTMR behavior Capture Timer counts until next unit period event 2h (R/W) = QPOSCNT behavior Position counter is unaffected by emulation suspend 2h (R/W) = QWDTMR behavior Watchdog counter is unaffected by emulation suspend 2h (R/W) = QUTMR behavior Unit timer is unaffected by emulation suspend 2h (R/W) = QCTMR behavior Capture Timer is unaffected by emulation suspend 3h (R/W) = Same as FREE_SOFT_2
13-12	PCRM	R/W	0h	Postion counter reset Reset type: SYSRSn 0h (R/W) = Position counter reset on an index event 1h (R/W) = Position counter reset on the maximum position 2h (R/W) = Position counter reset on the first index event 3h (R/W) = Position counter reset on a unit time event
11-10	SEI	R/W	0h	Strobe event initialization of position counter Reset type: SYSRSn 0h (R/W) = Does nothing (action disabled) 1h (R/W) = Does nothing (action disabled) 2h (R/W) = Initializes the position counter on rising edge of the QEPS signal 3h (R/W) = Clockwise Direction: Initializes the position counter on the rising edge of QEPS strobe Counter Clockwise Direction: Initializes the position counter on the falling edge of QEPS strobe

Table 17-19. QEPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	IEI	R/W	0h	Index event init of position count Reset type: SYSRSn 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Do nothing (action disabled) 2h (R/W) = Initializes the position counter on the rising edge of the QEPI signal (QPOSCNT = QPOSINIT) 3h (R/W) = Initializes the position counter on the falling edge of QEPI signal (QPOSCNT = QPOSINIT)
7	SWI	R/W	0h	Software init position counter Reset type: SYSRSn 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Initialize position counter (QPOSCNT=QPOSINIT). This bit is not cleared automatically
6	SEL	R/W	0h	Strobe event latch of position counter Reset type: SYSRSn 0h (R/W) = The position counter is latched on the rising edge of QEPS strobe (QPOSSLAT = POSCCNT). Latching on the falling edge can be done by inverting the strobe input using the QSP bit in the QDECCTL register 1h (R/W) = Clockwise Direction: Position counter is latched on rising edge of QEPS strobe Counter Clockwise Direction: Position counter is latched on falling edge of QEPS strobe
5-4	IEL	R/W	0h	Index event latch of position counter (software index marker) Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = Latches position counter on rising edge of the index signal 2h (R/W) = Latches position counter on falling edge of the index signal 3h (R/W) = Software index marker. Latches the position counter and quadrature direction flag on index event marker. The position counter is latched to the QPOSILAT register and the direction flag is latched in the QEPSTS[QDLF] bit. This mode is useful for software index marking.
3	QPEN	R/W	0h	Quadrature position counter enable/software reset Reset type: SYSRSn 0h (R/W) = Reset the eQEP peripheral internal operating flags/read-only registers. Control/configuration registers are not disturbed by a software reset. When QPEN is disabled, some flags in the QFLG register do not get reset or cleared and show the actual state of that flag. 1h (R/W) = eQEP position counter is enabled
2	QCLM	R/W	0h	QEP capture latch mode Reset type: SYSRSn 0h (R/W) = Latch on position counter read by CPU. Capture timer and capture period values are latched into QCTMRLAT and QCPRDLAT registers when CPU reads the QPOSCNT register. 1h (R/W) = Latch on unit time out. Position counter, capture timer and capture period values are latched into QPOSLAT, QCTMRLAT and QCPRDLAT registers on unit time out.
1	UTE	R/W	0h	QEP unit timer enable Reset type: SYSRSn 0h (R/W) = Disable eQEP unit timer 1h (R/W) = Enable unit timer
0	WDE	R/W	0h	QEP watchdog enable Reset type: SYSRSn 0h (R/W) = Disable the eQEP watchdog timer 1h (R/W) = Enable the eQEP watchdog timer

17.12.2.14 QCAPCTL Register (Offset = 16h) [Reset = 0000h]

QCAPCTL is shown in Figure 17-39 and described in Table 17-20.

Return to the Summary Table.

Qaudrature Capture Control

Figure 17-39. QCAPCTL Register

15	14	13	12	11	10	9	8
CEN	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED	CCPS			UPPS			
R-0h	R/W-0h			R/W-0h			

Table 17-20. QCAPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CEN	R/W	0h	Enable eQEP capture Reset type: SYSRSn 0h (R/W) = eQEP capture unit is disabled 1h (R/W) = eQEP capture unit is enabled
14-7	RESERVED	R	0h	Reserved
6-4	CCPS	R/W	0h	eQEP capture timer clock prescaler Reset type: SYSRSn 0h (R/W) = CAPCLK = SYSCLKOUT/1 1h (R/W) = CAPCLK = SYSCLKOUT/2 2h (R/W) = CAPCLK = SYSCLKOUT/4 3h (R/W) = CAPCLK = SYSCLKOUT/8 4h (R/W) = CAPCLK = SYSCLKOUT/16 5h (R/W) = CAPCLK = SYSCLKOUT/32 6h (R/W) = CAPCLK = SYSCLKOUT/64 7h (R/W) = CAPCLK = SYSCLKOUT/128
3-0	UPPS	R/W	0h	Unit position event prescaler Reset type: SYSRSn 0h (R/W) = UPEVNT = QCLK/1 1h (R/W) = UPEVNT = QCLK/2 2h (R/W) = UPEVNT = QCLK/4 3h (R/W) = UPEVNT = QCLK/8 4h (R/W) = UPEVNT = QCLK/16 5h (R/W) = UPEVNT = QCLK/32 6h (R/W) = UPEVNT = QCLK/64 7h (R/W) = UPEVNT = QCLK/128 8h (R/W) = UPEVNT = QCLK/256 9h (R/W) = UPEVNT = QCLK/512 Ah (R/W) = UPEVNT = QCLK/1024 Bh (R/W) = UPEVNT = QCLK/2048 Ch (R/W) = Reserved Dh (R/W) = Reserved Eh (R/W) = Reserved Fh (R/W) = Reserved

17.12.2.15 QPOSCTL Register (Offset = 17h) [Reset = 0000h]

QPOSCTL is shown in Figure 17-40 and described in Table 17-21.

Return to the Summary Table.

Position Compare Control

Figure 17-40. QPOSCTL Register

15	14	13	12	11	10	9	8
PCSHDW	PCLOAD	PCPOL	PCE	PCSPW			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			
7	6	5	4	3	2	1	0
PCSPW							
R/W-0h							

Table 17-21. QPOSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	PCSHDW	R/W	0h	Position compare of shadow enable Reset type: SYSRSn 0h (R/W) = Shadow disabled, load Immediate 1h (R/W) = Shadow enabled
14	PCLOAD	R/W	0h	Position compare of shadow load Reset type: SYSRSn 0h (R/W) = Load on QPOSCNT = 0 1h (R/W) = Load when QPOSCNT = QPOSCMP
13	PCPOL	R/W	0h	Polarity of sync output Reset type: SYSRSn 0h (R/W) = Active HIGH pulse output 1h (R/W) = Active LOW pulse output
12	PCE	R/W	0h	Position compare enable/disable Reset type: SYSRSn 0h (R/W) = Disable position compare unit 1h (R/W) = Enable position compare unit
11-0	PCSPW	R/W	0h	Select-position-compare sync output pulse width Reset type: SYSRSn 0h (R/W) = 1 * 4 * SYSCLKOUT cycles 1h (R/W) = 2 * 4 * SYSCLKOUT cycles FFFh (R/W) = 4096 * 4 * SYSCLKOUT cycles

17.12.2.16 QEINT Register (Offset = 18h) [Reset = 0000h]

QEINT is shown in Figure 17-41 and described in Table 17-22.

Return to the Summary Table.

QEP Interrupt Control

Figure 17-41. QEINT Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	QPE	PCE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 17-22. QEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R/W	0h	QMA Error Interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
11	UTO	R/W	0h	Unit time out interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
10	IEL	R/W	0h	Index event latch interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
9	SEL	R/W	0h	Strobe event latch interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
8	PCM	R/W	0h	Position-compare match interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
7	PCR	R/W	0h	Position-compare ready interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
6	PCO	R/W	0h	Position counter overflow interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
5	PCU	R/W	0h	Position counter underflow interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
4	WTO	R/W	0h	Watchdog time out interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled

Table 17-22. QEINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	QDC	R/W	0h	Quadrature direction change interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
2	QPE	R/W	0h	Quadrature phase error interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
1	PCE	R/W	0h	Position counter error interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
0	RESERVED	R	0h	Reserved

17.12.2.17 QFLG Register (Offset = 19h) [Reset = 0000h]

QFLG is shown in Figure 17-42 and described in Table 17-23.

Return to the Summary Table.

QEP Interrupt Flag

Figure 17-42. QFLG Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 17-23. QFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R	0h	QMA Error interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Interrupt was generated
11	UTO	R	0h	Unit time out interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Set by eQEP unit timer period match
10	IEL	R	0h	Index event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set after latching the QPOSCNT to QPOSILAT
9	SEL	R	0h	Strobe event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set after latching the QPOSCNT to QPOSSLAT
8	PCM	R	0h	eQEP compare match event interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set on position-compare match
7	PCR	R	0h	Position-compare ready interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set after transferring the shadow register value to the active position compare register
6	PCO	R	0h	Position counter overflow interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set on position counter overflow.
5	PCU	R	0h	Position counter underflow interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set on position counter underflow.
4	WTO	R	0h	Watchdog timeout interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Set by watchdog timeout

Table 17-23. QFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	QDC	R	0h	Quadrature direction change interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Interrupt was generated
2	PHE	R	0h	Quadrature phase error interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Set on simultaneous transition of QEPA and QEPB
1	PCE	R	0h	Position counter error interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Position counter error
0	INT	R	0h	Global interrupt status flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Interrupt was generated

17.12.2.18 QCLR Register (Offset = 1Ah) [Reset = 0000h]

QCLR is shown in Figure 17-43 and described in Table 17-24.

Return to the Summary Table.

QEP Interrupt Clear

Figure 17-43. QCLR Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 17-24. QCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R-0/W1S	0h	Clear QMA Error interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
11	UTO	R-0/W1S	0h	Clear unit time out interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
10	IEL	R-0/W1S	0h	Clear index event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
9	SEL	R-0/W1S	0h	Clear strobe event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
8	PCM	R-0/W1S	0h	Clear eQEP compare match event interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
7	PCR	R-0/W1S	0h	Clear position-compare ready interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
6	PCO	R-0/W1S	0h	Clear position counter overflow interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
5	PCU	R-0/W1S	0h	Clear position counter underflow interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
4	WTO	R-0/W1S	0h	Clear watchdog timeout interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag

Table 17-24. QCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	QDC	R-0/W1S	0h	Clear quadrature direction change interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
2	PHE	R-0/W1S	0h	Clear quadrature phase error interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
1	PCE	R-0/W1S	0h	Clear position counter error interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
0	INT	R-0/W1S	0h	Global interrupt clear flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag

17.12.2.19 QFRC Register (Offset = 1Bh) [Reset = 0000h]

QFRC is shown in Figure 17-44 and described in Table 17-25.

Return to the Summary Table.

QEP Interrupt Force

Figure 17-44. QFRC Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 17-25. QFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R/W	0h	Force QMA error interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
11	UTO	R/W	0h	Force unit time out interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
10	IEL	R/W	0h	Force index event latch interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
9	SEL	R/W	0h	Force strobe event latch interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
8	PCM	R/W	0h	Force position-compare match interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
7	PCR	R/W	0h	Force position-compare ready interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
6	PCO	R/W	0h	Force position counter overflow interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
5	PCU	R/W	0h	Force position counter underflow interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
4	WTO	R/W	0h	Force watchdog time out interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt

Table 17-25. QFRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	QDC	R/W	0h	Force quadrature direction change interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
2	PHE	R/W	0h	Force quadrature phase error interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
1	PCE	R/W	0h	Force position counter error interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
0	RESERVED	R	0h	Reserved

17.12.2.20 QEPSTS Register (Offset = 1Ch) [Reset = 0000h]

QEPSTS is shown in Figure 17-45 and described in Table 17-26.

Return to the Summary Table.

QEP Status

Figure 17-45. QEPSTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
UPEVNT	FIDF	QDF	QDLF	COEF	CDEF	FIMF	PCEF
R/W1C-0h	R-0h	R-0h	R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R-0h

Table 17-26. QEPSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	UPEVNT	R/W1C	0h	Unit position event flag Reset type: SYSRSn 0h (R/W) = No unit position event detected 1h (R/W) = Unit position event detected. Write 1 to clear
6	FIDF	R	0h	Direction on the first index marker Status of the direction is latched on the first index event marker. Reset type: SYSRSn 0h (R/W) = Counter-clockwise rotation (or reverse movement) on the first index event 1h (R/W) = Clockwise rotation (or forward movement) on the first index event
5	QDF	R	0h	Quadrature direction flag Reset type: SYSRSn 0h (R/W) = Counter-clockwise rotation (or reverse movement) 1h (R/W) = Clockwise rotation (or forward movement)
4	QDLF	R	0h	eQEP direction latch flag Reset type: SYSRSn 0h (R/W) = Counter-clockwise rotation (or reverse movement) on index event marker 1h (R/W) = Clockwise rotation (or forward movement) on index event marker
3	COEF	R/W1C	0h	Capture overflow error flag Reset type: SYSRSn 0h (R/W) = Overflow has not occurred. 1h (R/W) = Overflow occurred in eQEP Capture timer (QEPCTMR). This bit is cleared by writing a '1'.
2	CDEF	R/W1C	0h	Capture direction error flag Reset type: SYSRSn 0h (R/W) = Capture direction error has not occurred. 1h (R/W) = Direction change occurred between the capture position event. This bit is cleared by writing a '1'.
1	FIMF	R/W1C	0h	First index marker flag Reset type: SYSRSn 0h (R/W) = First index pulse has not occurred. 1h (R/W) = Set by first occurrence of index pulse. This bit is cleared by writing a '1'.

Table 17-26. QEPSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PCEF	R	0h	Position counter error flag. This bit is not sticky and it is updated for every index event. Reset type: SYSRSn 0h (R/W) = No error occurred during the last index transition 1h (R/W) = Position counter error

17.12.2.21 QCTMR Register (Offset = 1Dh) [Reset = 0000h]

QCTMR is shown in Figure 17-46 and described in Table 17-27.

Return to the Summary Table.

QEP Capture Timer

Figure 17-46. QCTMR Register

15	14	13	12	11	10	9	8
QCTMR							
R/W-0h							
7	6	5	4	3	2	1	0
QCTMR							
R/W-0h							

Table 17-27. QCTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCTMR	R/W	0h	This register provides time base for edge capture unit. Reset type: SYSRSn

17.12.2.22 QCPRD Register (Offset = 1Eh) [Reset = 0000h]

QCPRD is shown in Figure 17-47 and described in Table 17-28.

Return to the Summary Table.

QEP Capture Period

Figure 17-47. QCPRD Register

15	14	13	12	11	10	9	8
QCPRD							
R/W-0h							
7	6	5	4	3	2	1	0
QCPRD							
R/W-0h							

Table 17-28. QCPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCPRD	R/W	0h	This register holds the period count value between the last successive eQEP position events Reset type: SYSRSn

17.12.2.23 QCTMRLAT Register (Offset = 1Fh) [Reset = 0000h]

QCTMRLAT is shown in Figure 17-48 and described in Table 17-29.

Return to the Summary Table.

QEP Capture Latch

Figure 17-48. QCTMRLAT Register

15	14	13	12	11	10	9	8
QCTMRLAT							
R-0h							
7	6	5	4	3	2	1	0
QCTMRLAT							
R-0h							

Table 17-29. QCTMRLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCTMRLAT	R	0h	The eQEP capture timer value can be latched into this register on two events viz., unit timeout event, reading the eQEP position counter. Reset type: SYSRSn

17.12.2.24 QCPRDLAT Register (Offset = 20h) [Reset = 0000h]

QCPRDLAT is shown in Figure 17-49 and described in Table 17-30.

Return to the Summary Table.

QEP Capture Period Latch

Figure 17-49. QCPRDLAT Register

15	14	13	12	11	10	9	8
QCPRDLAT							
R-0h							
7	6	5	4	3	2	1	0
QCPRDLAT							
R-0h							

Table 17-30. QCPRDLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCPRDLAT	R	0h	eQEP capture period value can be latched into this register on two events viz., unit timeout event, reading the eQEP position counter. Reset type: SYSRSn

17.12.2.25 REV Register (Offset = 30h) [Reset = 00000009h]

REV is shown in Figure 17-50 and described in Table 17-31.

Return to the Summary Table.

QEP Revision Number

Figure 17-50. REV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										MINOR			MAJOR		
R-0-0h										R-1h			R-1h		

Table 17-31. REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-3	MINOR	R	1h	This field specifies the Minor Revision number for the eQEP IP. Reset type: N/A
2-0	MAJOR	R	1h	This field specifies the Major Revision number for the eQEP IP. Reset type: N/A

17.12.2.26 QEPSTROBESEL Register (Offset = 32h) [Reset = 00000000h]

QEPSTROBESEL is shown in Figure 17-51 and described in Table 17-32.

Return to the Summary Table.

QEP Strobe select register

Figure 17-51. QEPSTROBESEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						STROBESEL	
R-0-0h						R/W-0h	

Table 17-32. QEPSTROBESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1-0	STROBESEL	R/W	0h	Strobe source select: Reset type: SYSRSn 0h (R/W) = QEP Strobe after polarity mux 1h (R/W) = QEP Strobe after polarity mux 2h (R/W) = QEP Strobe after polarity mux ORed with ADCSOCA 3h (R/W) = QEP Strobe after polarity mux ORed with ADCSOCB

17.12.2.27 QMACTRL Register (Offset = 34h) [Reset = 00000000h]

QMACTRL is shown in Figure 17-52 and described in Table 17-33.

Return to the Summary Table.

QMA Control register

Figure 17-52. QMACTRL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													MODE		
R-0-0h													R/W-0h		

Table 17-33. QMACTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2-0	MODE	R/W	0h	Select Mode for QMA mode: 000 : QMA Module is bypassed. 001 : QMA Mode-1 operation selected 010 : QMA Mode-2 operation selected 011 : QMA Module is bypassed (reserved) 1xx : QMA Module is bypassed (reserved) Reset type: SYSRSn

17.12.2.28 QEPSRCSEL Register (Offset = 36h) [Reset = 00000000h]

QEPSRCSEL is shown in Figure 17-53 and described in Table 17-34.

Return to the Summary Table.

QEP Source Select Register

Figure 17-53. QEPSRCSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QEPSSEL				QEPISEL				QEPBSEL				QEPASEL			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

Table 17-34. QEPSRCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-12	QEPSSEL	R/W	0h	QEP Strobe source select: 0x0: From device Pins (Default). Others: Tied to zero. Reset type: SYSRSn
11-8	QEPISEL	R/W	0h	QEP Index source select: 0x0: Device Pin (Default) 0x1 to 0x1F : Refer to EQEP chaper of TRM Note: eQEP needs to be disabled before configuring these bits as it can lead to unexpected behavior if eQEP is running. Reset type: SYSRSn
7-4	QEPBSEL	R/W	0h	QEPB source select: 0x0: Device Pin (Default) 0x1 to 0x1F : Refer to EQEP chaper of TRM Note: eQEP needs to be disabled before configuring these bits as it can lead to unexpected behavior if eQEP is running. Reset type: SYSRSn
3-0	QEPASEL	R/W	0h	QEPA source select: 0x0: Device Pin (Default) 0x1 to 0x1F : Refer to EQEP chaper of TRM Note: eQEP needs to be disabled before configuring these bits as it can lead to unexpected behavior if eQEP is running. Reset type: SYSRSn

Chapter 18

Universal Asynchronous Receiver/Transmitter (UART)

This chapter describes the Universal Asynchronous Receivers/Transmitters (UARTs).

18.1 Introduction

The UART module performs the functions of parallel-to-serial and serial-to-parallel conversions.

18.1.1 Features

The Universal Asynchronous Receiver/Transmitter (UART) module in this device contains the following features:

- Programmable baud-rate generator allowing speeds up to Mbps for regular speed (divide by 16) and Mbps for high speed (divide by 8)
- Separate 16-deep and 8-bit wide transmit (TX) and receive (RX) FIFOs to reduce CPU interrupt service loading
- Programmable FIFO length providing conventional double-buffered interface
- FIFO trigger levels of ⅛, ¼, ½, ¾, and ⅞ in regular FIFO level mode, and 1/16, 2/16, 3/16, 4/16 … 15/16, 16/16 in fine-grained FIFO level mode
- Standard asynchronous communication bits for start, stop, and parity
- Line-break generation and detection
- Fully programmable serial interface characteristics
  - 5, 6, 7, or 8 data bits
  - Even, odd, stick, or no parity bit generation and detection
  - 1 or 2 stop bit generation
- IrDA serial-IR (SIR) encoder and decoder providing
  - Programmable use of IrDA SIR or UART input/output
  - Support of IrDA SIR encoder and decoder functions for data rates up to 115.2kbps half-duplex
  - Support of normal 3/16 and low-power (1.41 to 2.23µs) bit durations
  - Programmable internal clock generator enabling division of reference clock by 1 to 256 for low-power mode bit duration
- EIA-485 9-bit support
- Standard FIFO-level and End-of-Transmission (EOT) interrupts
- Efficient transfers using Direct Memory Access Controller (DMA)
  - Separate channels for transmit and receive
  - Receive single request asserted when data is in the FIFO; burst request asserted at programmed FIFO level
  - Transmit single request asserted when there is space in the FIFO; burst request asserted at programmed FIFO level
- SYSCLK is used to generate the baud clock.

18.1.2 Block Diagram

Figure 18-1 shows the UART block diagram.

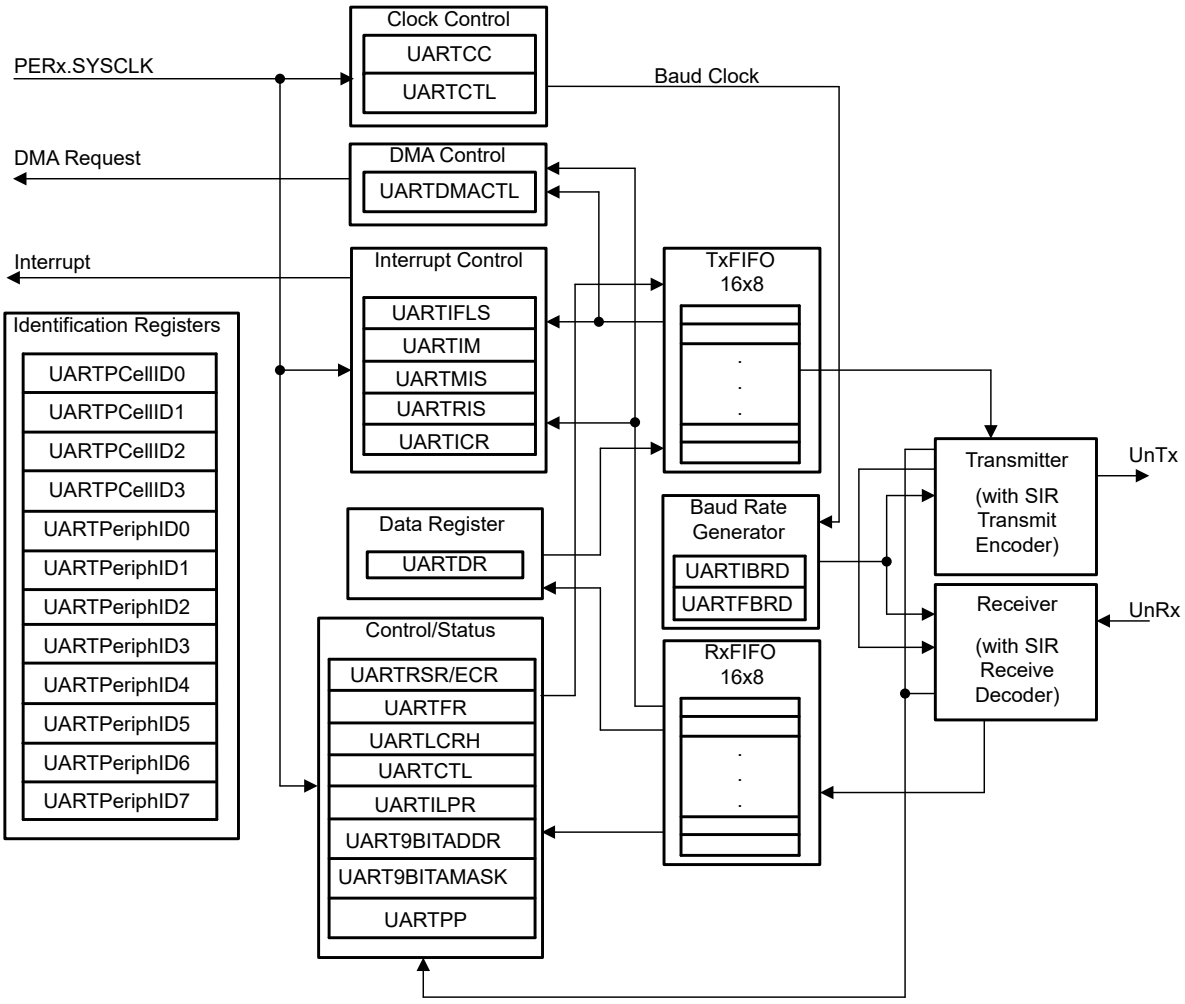


Figure 18-1. UART Module Block Diagram

18.2 Functional Description

The UART module performs the functions of parallel-to-serial and serial-to-parallel conversions. The UART module is similar in functionality to the SCI module (found on some C2000 devices), but is not directly register-compatible.

The UART is configured for transmit or receive through the TXE and RXE bits of the UART Control (UARTCTL) register. Transmit and receive are both enabled out of reset. Before any control registers are programmed, the UART must be disabled by clearing the UARTEN bit in UARTCTL. If the UART is disabled during a TX or RX operation, the current transaction is completed prior to the UART stopping.

The UART module also includes a serial IR (SIR) encoder and decoder block that can be connected to an infrared transceiver to implement an IrDA SIR physical layer. The SIR function is programmed using the UARTCTL register.

18.2.1 Transmit and Receive Logic

The transmit logic performs parallel-to-serial conversion on the data read from the transmit FIFO. The control logic outputs the serial bit stream beginning with a start bit and followed by the data bits (LSB first), parity bit, and the stop bits according to the programmed configuration in the control registers. See Figure 18-2 for details.

The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line-break detection are also performed, and the status accompanies the data that is written to the receive FIFO.

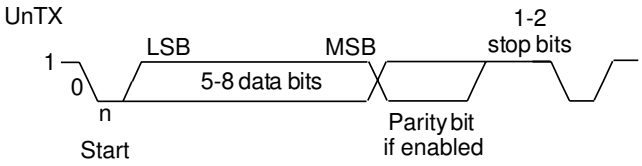


Figure 18-2. UART Character Frame

18.2.2 Baud-Rate Generation

The baud-rate divisor is a 22-bit number consisting of a 16-bit integer and a 6-bit fractional part. The number formed by these two values is used by the baud-rate generator to determine the bit period. Having a fractional baud-rate divisor allows the UART to generate all the standard baud rates.

The 16-bit integer is loaded through the UART Integer Baud-Rate Divisor (UARTIBRD) register and the 6-bit fractional part is loaded with the UART Fractional Baud-Rate Divisor (UARTFBRD) register. The baud-rate divisor (BRD) has the following relationship to the system clock (where BRDI is the integer part of the BRD, and BRDF is the fractional part, separated by a decimal place.)

$$BRD = BRDI + BRDF = UARTSysClk / (ClkDiv \times Baud\ Rate)$$

where

UARTSysClk is the system clock (SYSCLK) connected to the UART, and ClkDiv is either 16 (if HSE in UARTCTL is clear) or 8 (if HSE in UARTCTL is set).

By default, this is the main system clock (SYSCLK).

The 6-bit fractional number (that is to be loaded into the DIVFRAC bit field in the UARTFBRD register) can be calculated by taking the fractional part of the baud-rate divisor, multiplying by 64, and adding 0.5 to account for rounding errors:

$$UARTFBRD[DIVFRAC] = integer(BRDF \times 64 + 0.5)$$

The UART generates an internal baud-rate reference clock at 8× or 16× the baud-rate (referred to as Baud8 and Baud16, depending on the setting of the HSE bit [bit 5] in UARTCTL). This reference clock is divided by 8 or 16 to generate the transmit clock, and is used for error detection during receive operations.

Along with the UART Line Control High Byte (UARTLCRH) register, the UARTIBRD and UARTFBRD registers form an internal 30-bit register. This internal register is only updated when a write operation to UARTLCRH is performed, so any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register for the changes to take effect.

To update the baud-rate registers, there are four possible sequences:

- UARTIBRD write, UARTFBRD write, and UARTLCRH write
- UARTFBRD write, UARTIBRD write, and UARTLCRH write
- UARTIBRD write and UARTLCRH write
- UARTFBRD write and UARTLCRH write

18.2.3 Data Transmission

Data received or transmitted is stored in two 16-byte FIFOs, though the receive FIFO has an extra four bits per character for status information. For transmission, data is written into the transmit FIFO. If the UART is enabled, the UART causes a data frame to start transmitting with the parameters indicated in the UARTLCRH register. Data continues to be transmitted until there is no data left in the transmit FIFO. The BUSY bit in the UART Flag (UARTFR) register is asserted as soon as data is written to the transmit FIFO (that is, if the FIFO is non-empty) and remains asserted while data is being transmitted. The BUSY bit is negated only when the transmit FIFO is empty, and the last character has been transmitted from the shift register, including the stop bits. The UART can indicate that the UART is busy even though the UART is no longer enabled.

When the receiver is idle (the UnRx signal is continuously 1), and the data input goes Low (a start bit has been received), the receive counter begins running and data is sampled on the eighth cycle of Baud16 or fourth cycle of Baud8 depending on the setting of the HSE bit (bit 5) in UARTCTL (described in Section 18.2.1).

The start bit is valid and recognized if the UnRx signal is still low on the eighth cycle of Baud16 (HSE clear) or the fourth cycle of Baud8 (HSE set), otherwise the start bit is ignored. After a valid start bit is detected, successive data bits are sampled on every 16th cycle of Baud16 or eighth cycle of Baud8 (that is, one bit period later) according to the programmed length of the data characters and value of the HSE bit in UARTCTL. The parity bit is then checked if parity mode is enabled. Data length and parity are defined in the UARTLCRH register.

Lastly, a valid stop bit is confirmed if the UnRx signal is High, otherwise a framing error has occurred. When a full word is received, the data is stored in the receive FIFO along with any error bits associated with that word.

18.2.4 Serial IR (SIR)

The UART peripheral includes an IrDA SIR encoder and decoder block. The IrDA SIR block provides functionality that converts between an asynchronous UART data stream and a half-duplex serial SIR interface. No analog processing is performed on-chip. The role of the SIR block is to provide a digital encoded output and decoded input to the UART. When enabled, the SIR block uses the UnTx and UnRx pins for the SIR protocol. These signals must be connected to an infrared transceiver to implement an IrDA SIR physical layer link. The SIR block can receive and transmit, but the SIR block is only half-duplex so the SIR block cannot do both at the same time. Transmission must be stopped before data can be received. The IrDA SIR physical layer specifies a minimum 10ms delay between transmission and reception. The SIR block has two modes of operation:

- In normal IrDA mode, a zero logic level is transmitted as a high pulse of 3/16th duration of the selected baud rate bit period on the output pin, while logic one levels are transmitted as a static low signal. These levels control the driver of an infrared transmitter, sending a pulse of light for each zero. On the reception side, the incoming light pulses energize the photo transistor base of the receiver, pulling the output low and driving the UART input pin low.
- In low-power IrDA mode, the width of the transmitted infrared pulse is set to three times the period of the internally generated IrLPBaud16 signal (1.63µs, assuming a nominal 1.8432MHz frequency) by changing the appropriate bit in the UARTCTL register.

Whether the device is in normal or low-power IrDA mode, a start bit is deemed valid if the decoder is still low one period of IrLPBaud16 after the low was first detected. This enables a normal-mode UART to receive data from a low-power mode UART that can transmit pulses as small as 1.41µs. Thus, for both low-power and normal mode operation, the ILPDVSR field in the UARTILPR register must be programmed such that 1.42MHz < $f_{IrLPBaud16}$< 2.12MHz, resulting in a low-power pulse duration of 1.41 to 2.11µs (three times the period of IrLPBaud16). The minimum frequency of IrLPBaud16 make sure that pulses less than one period of IrLPBaud16 are rejected, but pulses greater than 1.4µs are accepted as valid pulses.

Figure 18-3 shows the UART transmit and receive signals, with and without IrDA modulation.

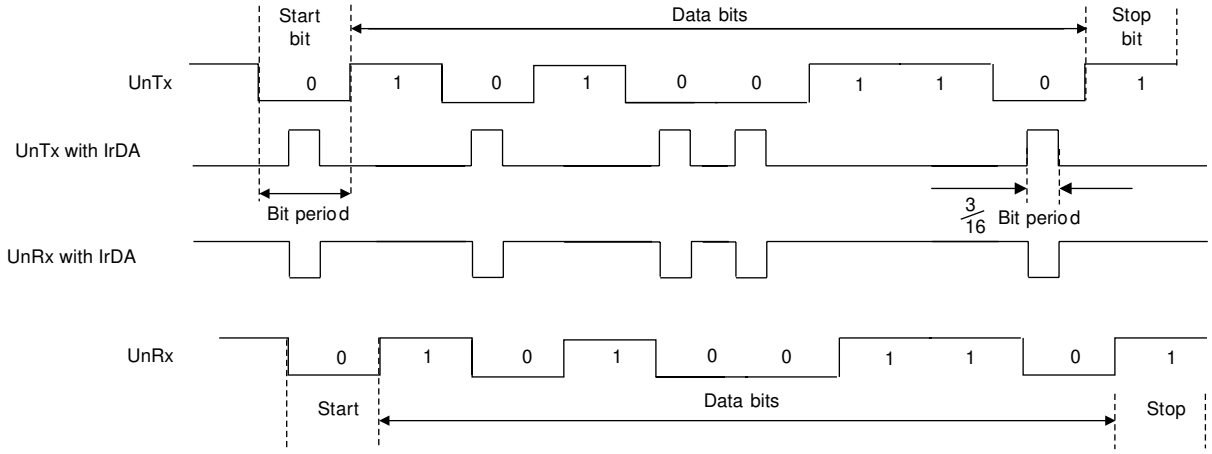


Figure 18-3. IrDA Data Modulation

In both normal and low-power IrDA modes:

- During transmission, the UART data bit is used as the base for encoding
- During reception, the decoded bits are transferred to the UART receive logic

The IrDA SIR physical layer specifies a half-duplex communication link, with a minimum 10ms delay between transmission and reception. This delay must be generated by software because the delay is not automatically supported by the UART. The delay is required because the infrared receiver electronics can become biased or even saturated from the optical power coupled from the adjacent transmitter LED. This delay is known as latency or receiver setup time.

18.2.5 9-Bit UART Mode

The UART provides a 9-bit mode that is enabled with the 9BITEN bit in the UART9BITADDR register. This feature is useful in a multi-drop configuration of the UART, where a single transmitter connected to multiple receivers can communicate with a particular receiver through the address or set of addresses along with a qualifier for an address byte. All the receivers check for the address qualifier in the place of the parity bit and, if set, then compare the byte received with the preprogrammed address. If the address matches, then the receiver receives or sends further data. If the address does not match, the receiver drops the address byte and any subsequent data bytes. If the UART is in 9-bit mode, then the receiver operates with no parity mode. The address can be predefined to match with the received byte and the address can be configured with the UART9BITADDR register. The matching can be extended to a set of addresses using the address mask in the UART9BITAMASK register. By default, the UART9BITAMASK is 0xFF, meaning that only the specified address is matched.

When not finding a match, the rest of the data bytes with the ninth bit cleared are dropped. If a match is found, then an interrupt is generated to the NVIC for further action. The subsequent data bytes with the cleared ninth bit are stored in the FIFO. Software can mask this interrupt in case DMA or FIFO operations are enabled for this instance and processor intervention is not required. All the send transactions with 9-bit mode are data bytes and the ninth bit is cleared. Software can override the ninth bit to be set (to indicate address) by overriding the parity settings to sticky parity with odd parity enabled for a particular byte. To match the transmission time with correct parity settings, the address byte can be transmitted as a single then a burst transfer. The transmit FIFO does not hold the address/data bit; hence, software can enable the address bit appropriately.

18.2.6 FIFO Operation

The UART has two 16-deep 8-bit wide FIFOs; one for transmit and one for receive. Both FIFOs are accessed by way of the UART Data (UARTDR) register. Read operations of the UARTDR register return a 12-bit value consisting of eight data bits and four error flags while write operations place 8-bit data in the transmit FIFO.

Out of reset, both FIFOs are disabled and act as 1 byte-deep holding registers. The FIFOs are enabled by setting the FEN bit in the UARTLCRH register.

FIFO status can be monitored through the UART Flag (UARTFR) register and the UART Receive Status (UARTRSR) register. Hardware monitors empty, full, and overrun conditions. The UARTFR register contains empty and full flags (TXFE, TXFF, RXFE, and RXFF bits), and the UARTRSR register shows overrun status with the OE bit. If the FIFOs are disabled, the empty and full flags are set according to the status of the 1 byte-deep holding registers. For CCS debugging, use the RXFIFOLVL and TXFIFOLVL fields to monitor how many characters are present in each FIFO.

The trigger points, at which the FIFOs generate interrupts, is controlled by way of the UART Interrupt FIFO Level Select (UARTIFLS) register. Both FIFOs can be individually configured to trigger interrupts at different levels. Available configurations include ⅛, ¼, ½, ¾, and ⅞ in the TXIFSEL and RXIFSEL fields of the UARTIFLS register. If single-byte FIFO level granularity is needed, set the TXIFSEL and RXIFSEL fields to the "fine-grained" setting. This enables the use of the TXIFSEL_FINE and RXIFSEL_FINE fields in the UARTIFLS register. These can instead be configured to trigger at any FIFO level between 0/16 and 15/16 for the TX FIFO and between 1/16 and 16/16 for the RX FIFO.

For example, if the ¼ option is selected for the receive FIFO, the UART generates a receive interrupt after four data bytes are received. If the ¼ option is selected for the transmit FIFO, the UART generates a transmit interrupt after four data bytes can be transmitted, meaning there are at least four empty spaces in the transmit FIFO. Out of reset, both FIFOs are configured to trigger an interrupt at the ½ mark.

18.2.7 Interrupts

The UART can generate interrupts when the following conditions are observed:

- Overrun error
- Break error
- Parity error
- Framing error
- Receive time-out
- Transmit (when condition defined in the TXIFLSEL bit in the UARTIFLS register is met, or if the EOT bit in UARTCTL is set, when the last bit of all transmitted data leaves the serializer)
- Receive (when condition defined in the RXIFLSEL bit in the UARTIFLS register is met)

All of the interrupt events are ORed together before being sent to the interrupt controller, so the UART can only generate a single interrupt request to the controller at any given time. Software can service multiple interrupt events in a single interrupt service routine by reading the UART Masked Interrupt Status (UARTMIS) register.

The interrupt events that can trigger a controller-level interrupt are defined in the UART Interrupt Mask (UARTIM) register by setting the corresponding IM bits. If interrupts are not used, the raw interrupt status is visible by way of the UART Raw Interrupt Status (UARTRIS) register.

Note

For receive time-out, the RTIM bit in the UARTIM register must be set to see the RTMIS and RTRIS status in the UARTMIS and UARTRIS registers.

Interrupts are always cleared (for the UARTMIS and UARTRIS registers) by writing a 1 to the corresponding bit in the UART Interrupt Clear (UARTICR) register. Additionally, write a 1 to the INT_FLG_CLR field of the UART_GLB_INT_CLR register to clear the global interrupt flag and receive further UART interrupts.

The receive time-out interrupt is asserted when the receive FIFO is not empty, and no further data is received over a 32-bit period when the HSE bit is clear or over a 64-bit period when the HSE bit is set. The receive time-out interrupt is cleared either when the FIFO becomes empty through reading all the data (or by reading the holding register), or when a 1 is written to the corresponding bit in the UARTICR register.

The receive interrupt changes state when one of the following events occurs:

- If the FIFOs are enabled and the receive FIFO reaches the programmed trigger level, the RXRIS bit is set. The receive interrupt is cleared by reading data from the receive FIFO until the receive FIFO becomes less than the trigger level, or by clearing the interrupt by writing a 1 to the RXIC bit.
- If the FIFOs are disabled (have a depth of one location) and data is received thereby filling the location, the RXRIS bit is set. The receive interrupt is cleared by performing a single read of the receive FIFO, or by clearing the interrupt by writing a 1 to the RXIC bit.

The transmit interrupt changes state when one of the following events occurs:

- If the FIFOs are enabled and the transmit FIFO progresses through the programmed trigger level, the TXRIS bit is set. The transmit interrupt is based on a transition through level, therefore the FIFO must be written past the programmed trigger level otherwise no further transmit interrupts are generated. The transmit interrupt is cleared by writing data to the transmit FIFO until the transmit FIFO becomes greater than the trigger level, or by clearing the interrupt by writing a 1 to the TXIC bit.
- If the FIFOs are disabled (have a depth of one location) and there is no data present in the transmitters single location, the TXRIS bit is set. The transmit interrupt is cleared by performing a single write to the transmit FIFO, or by clearing the interrupt by writing a 1 to the TXIC bit.

18.2.8 Loopback Operation

The UART can be placed into an internal loopback mode for diagnostic or debug work by setting the LBE bit in the UARTCTL register. In loopback mode, data transmitted on the UnTx output is received on the UnRx input. Note that the LBE bit must be set before the UART is enabled.

18.2.9 DMA Operation

The UART provides an interface to the DMA controller with separate channels for transmit and receive. The DMA operation of the UART is enabled through the UART DMA Control (UARTDMACTL) register. When DMA operation is enabled, the UART asserts a DMA request on the receive or transmit channel when the associated FIFO can transfer data. To trigger the DMA with the UART trigger sources, the UART FIFOs on the corresponding channel must be enabled. DMA operation with UART does not work in non-FIFO mode.

The UARTx_TX and UARTx_RX DMA settings will trigger the DMA on the FIFO level condition. For the receive channel, a burst request is asserted whenever the amount of data in the receive FIFO is at or above the FIFO trigger level configured in the UARTIFLS register. For the transmit channel, a burst request is asserted whenever the transmit FIFO contains equal or fewer characters than the FIFO trigger level. The UARTx_TX_SREQ and UARTx_RX_SREQ settings will trigger the DMA on a single request, meaning there is at least one element empty in the TX FIFO or at least one element full in the RX FIFO. The burst DMA transfer requests are handled automatically by the DMA controller depending on how the DMA channel is configured. When using the DMA to transfer 16-bit or 32-bit data to UARTDR for a transmit, only the 8 least-significant bits are transmitted.

To enable DMA operation for the receive channel, set the RXDMAE bit of the DMA Control (UARTDMACTL) register. To enable DMA operation for the transmit channel, set the TXDMAE bit of the UARTDMACTL register. The UART can also be configured to stop using DMA for the receive channel if a receive error occurs. If the DMAERR bit of the UARTDMACR register is set and a receive error occurs, the DMA receive requests are automatically disabled. This error condition can be cleared by clearing the appropriate UART error interrupt.

When the DMA controller is finished transferring data to the TX FIFO or from the RX FIFO, a dma_done signal is sent to the UART to indicate completion. The dma_done status is indicated through the DMATXRIS and DMARXIS bits of the UARTRIS register. An interrupt can be generated from these status bits by setting the DMATXIM and DMARXIM bits in the UARTIM register.

Note

The DMATXRIS bit can be used to indicate the completion of data transfer from the DMA controller to the TX FIFO. To indicate transfer completion from the serializer of the UART, the EOT bit can be enabled in the UARTCTL register.

See the *Direct Memory Access (DMA)* chapter for more details about programming the DMA controller.

18.2.9.1 Receiving Data Using UART with DMA

When using the DMA with the RX FIFO, the DMA Burst Size [DMA_BURST_SIZE] must equal the number of bytes used to trigger the FIFO level interrupt defined by RXIFLSEL in UARTIFLS. For example, a level of 1/8 is triggered by 2 bytes or more present in the FIFO, so DMA_BURST_SIZE = 2. To make sure that the DMA correctly receives all data from the RX FIFO, the DMA Burst Size also must be an integer divisor of the total number of UART receives. For complete data reception, follow these steps:

1. Decide the total number of words to be received. [BUFFER_SIZE]
2. Decide the necessary FIFO level [UART_BUFFER_SIZE] and set RXIFSEL accordingly.
3. Calculate the number of DMA transfers. [DMA_TRANSFER_SIZE = BUFFER_SIZE / UART_BUFFER_SIZE]
4. Set the DMA Burst Size. [DMA_BURST_SIZE] equal to the UART_BUFFER_SIZE.
5. Configure DMA using the calculated values and use the UART peripheral trigger as the trigger source.

To receive 120 words from the UART using the DMA:

BUFFER_SIZE = 120

UART_BUFFER_SIZE = 12

RXIFSEL: UART_BUFFER_SIZE / 16 = 3/4

DMA_TRANSFER_SIZE: BUFFER_SIZE / UART_BUFFER-SIZE = 120/12 = 10

DMA_BURST_SIZE: UART_BUFFER_SIZE = 12

(1 burst = 12 words, 1 transfer = 10 bursts, 120 words received with each transfer)

18.2.9.2 Transmitting Data Using UART with DMA

When using the DMA with the TX FIFO, the DMA Burst Size [DMA_BURST_SIZE] must equal 16 - the number of bytes used to trigger the FIFO level interrupt defined by TXIFSEL in UARTIFLS. For example, a level of 1/8 is triggered by having 14 bytes or less present in the FIFO, so DMA_BURST_SIZE = 16 - 14 = 2. To make sure that the DMA writes all data correctly to the TX FIFO, the DMA Burst Size must also be an integer divisor of the total number of UART transmissions. For complete data transmission, follow these steps:

1. Decide the total number or words to be transmitted. [BUFFER_SIZE]
2. Decide the necessary FIFO level [UART_BUFFER_SIZE] and set TXIFSEL accordingly.
3. Calculate the number of DMA transfers. [DMA_TRANSFER_SIZE = BUFFER_SIZE / UART_BUFFER_SIZE]
4. Calculate the DMA Burst Size. [DMA_BURST_SIZE] to equal 16 - UART_BUFFER_SIZE.
5. Configure the DMA using the calculated values and use UART peripheral trigger as the trigger source.

Note

For a BUFFER_SIZE less than or equal to 16, a software trigger can alternatively be used.

To transmit 120 words from the UART using the DMA:

BUFFER_SIZE = 120

UART_BUFFER_SIZE = 12

TXIFSEL: 1 - (UART_BUFFER_SIZE / 16) = 1 - 3/4 = 1/4

DMA_TRANSFER_SIZE: (BUFFER_SIZE / UART_BUFFER-SIZE) = (120/12) = 10

DMA_BURST_SIZE: UART_BUFFER_SIZE = 12

(1 burst = 12 words, 1 transfer = 10 bursts, 120 words transmitted with each transfer)

18.3 Initialization and Configuration

To enable and initialize the UART, perform the following steps:

1. Disable the UART by clearing the UARTEN bit in the UARTCTL register.
2. Write the integer portion of the BRD to the UARTIBRD register.
3. Write the fractional portion of the BRD to the UARTFBRD register.
4. Write the desired serial parameters to the UARTLCRH register.
5. Optionally, configure the DMA channel (see the *Direct Memory Access (DMA)* chapter and enable the DMA options in the UARTDMACTL register.
6. Enable the UART by setting the UARTEN bit in the UARTCTL register.

18.4 Software

18.4.1 UART Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/uart

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

18.4.1.1 UART Echoback

FILE: uart_ex1_echoback.c

This test receives and echo-backs data through the UARTA port.

A terminal such as 'putty' can be used to view the data from the UART and to send information to the UART. Characters received by the UART port are sent back to the host.

Running the Application Open a COM port with the following settings using a terminal:
- Find correct COM port
- Bits per second = 115200
- Data Bits = 8
- Parity = None
- Stop Bits = 1
- Hardware Control = None

The program will print out a greeting and then ask you to enter a character which it will echo back to the terminal.

Watch Variables
- None

External Connections
Connect the UARTA port to a PC via a transceiver and cable.
- GPIO32 is UARTARX (Connect to Pin3, PC-TX, of serial DB9 cable)
- GPIO43 is UARTATX (Connect to Pin2, PC-RX, of serial DB9 cable)

The pin muxing for the UARTA port needs to be done by the master CPU1. The common configuration example provided in the C28x folder can be used for assigning appropriate GPIO pins for UART RX and TX.

18.4.1.2 UART Loopback

FILE: uart_ex2_loopback.c

This test receives and loop-backs data through the UARTA port.

Watch Variables
- *sData* - Data being sent
- *rData* - Data received
- *loopCount* - Number of characters sent
- *errorCount* - Number of errors detected

Note: Avoid keeping the memory browser open while the execution is in progress.

External Connections
Connect the UARTA Rx port to Tx port for external loopback.
- GPIO32 is UARTARX to be connected to GPIO43 is UARTATX

18.4.1.3 UART Loopback with interrupt

FILE: uart_ex3_loopback_interrupt.c

This test uses the internal loop back test mode of the peripheral. Receive interrupt with FIFO is used in the example.

A stream of data is sent and then compared to the received stream. The UART-A sent data looks like this:
00 01
01 02
02 03
....
FE FF
FF 00
etc..
The pattern is repeated forever.

Watch Variables

- *sData* - Data being sent
- *rData* - Data received
- *loopCount* - Number of times ISR is executed
- *errorCount* - Number of errors detected

Note: Avoid keeping the memory browser open while the execution is in progress.

External Connections
Connect the UARTA Rx port to Tx port for external loopback.

- GPIO32 is UARTARX to be connected to GPIO43 is UARTATX

18.4.1.4 UART Digital Loopback with DMA

FILE: uart_ex4_loopback_dma.c

This program uses the internal loopback test mode of the UART module. Two DMA interrupts and both of the UART FIFOs are used. Both DMA channels are configured to transfer BUFFER_SIZE words with each burst being UART_BUFFER_SIZE words. The configurations for the UART module and both DMA modules are done in the Sysconfig file.

Note: Only the lower 8 bits of each word will be transmitted and received by the UART since the UART data register only has an 8 bit data field.

When the UART transmit FIFO has enough space (UART_BUFFER_SIZE bytes empty), the TX DMA will transfer data from the txData buffer in GSRAM memory into the transmit FIFO by writing to the UART data register. The data in the transmit FIFO will then be transmitted to the receive FIFO via the internal loopback.

When enough data has been placed in the receive FIFO (UART_BUFFER_SIZE data bytes full), the RX DMA will transfer the data from the receive FIFO into the rxData buffer in GSRAM memory by reading the UART data register.

When BUFFER_SIZE words have been placed into rxData, a check of the validity of the data will be performed and the errCount variable will indicate the amount of data mismatches between txData and rxData.

Running the Application

Set BUFFER_SIZE to desired words of data to transfer. This value must be even numbered. Set UART_BUFFER_SIZE to 2, 4, 8, 12 or 14. This value must divide evenly into the chosen BUFFER_SIZE. Change "Burst Size" and "Transfer Size" settings in Sysconfig (for both myDMA0 and myDMA1) to reflect these values. Burst Size = UART_BUFFER_SIZE Transfer Size = BUFFER_SIZE / UART_BUFFER_SIZE

To use 32-bit words instead of 16-bit words, change:

- txData and rxData data types to uint32_t.
- "Databus Width" setting in Sysconfig (for both myDMA0 and myDMA1) to "DMA transfers 32 bits at a time."
- DMA0 "Source Address Burst Step" and "Source Address Transfer Step" in Sysconfig to 2.
- DMA1 "Destination Address Burst Step" and "Destination Address Transfer Step" in Sysconfig to 2.

NOTE: This example implements the UART FIFO Buffers in NON fine-grained mode

External Connections (Optional)

- Connect UARTA_TX and UARTA_RX for external loopback. Uncheck "Enable Loopback Mode" in Sysconfig myUART0 instance
- UARTA_TX is on GPIO43
- UARTA_RX is on GPIO32

Watch Variables

- *txData* - Data transmitted
- *rxData* - Data received
- *errCount* - Error count

18.5 UART Registers

This Section describes the UART Registers.

18.5.1 UART Base Address Table

Table 18-1. UART Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
UartaRegs, UartaWriteRegs	UART_REGS , UART_REGS_WRITE	UARTA_BASE, UARTAWRITE_BASE	0x0006_A000	YES	YES

18.5.2 UART_REGS Registers

Table 18-2 lists the memory-mapped registers for the UART_REGS registers. All register offset addresses not listed in Table 18-2 should be considered as reserved locations and the register contents should not be modified.

Table 18-2. UART_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	UARTDR	UART Data	
2h	UARTRSR	UART Receive Status/Error Clear	
Ch	UARTFR	UART Flag	
10h	UARTILPR	UART IrDA Low-Power Register	
12h	UARTIBRD	UART Integer Baud-Rate Divisor	
14h	UARTFBRD	UART Fractional Baud-Rate Divisor	
16h	UARTLCRH	UART Line Control	
18h	UARTCTL	UART Control	
1Ah	UARTIFLS	UART Interrupt FIFO Level Select	
1Ch	UARTIM	UART Interrupt Mask	
1Eh	UARTRIS	UART Raw Interrupt Status	
20h	UARTMIS	UART Masked Interrupt Status	
22h	UARTICR	UART Interrupt Clear	
24h	UARTDMACTL	UART DMA Control	
40h	UART_GLB_INT_EN	UART Global Interrupt Enable Register	
42h	UART_GLB_INT_FLG	UART Global Interrupt Flag Register	
44h	UART_GLB_INT_CLR	UART Global Interrupt Clear Register	
52h	UART9BITADDR	UART 9-Bit Self Address	
54h	UART9BITAMASK	UART 9-Bit Self Address Mask	
7E0h	UARTPP	UART Peripheral Properties	
7E8h	UARTPeriphID4	UART Peripheral Identification 4	
7EAh	UARTPeriphID5	UART Peripheral Identification 5	
7ECh	UARTPeriphID6	UART Peripheral Identification 6	
7EEh	UARTPeriphID7	UART Peripheral Identification 7	
7F0h	UARTPeriphID0	UART Peripheral Identification 0	
7F2h	UARTPeriphID1	UART Peripheral Identification 1	
7F4h	UARTPeriphID2	UART Peripheral Identification 2	
7F6h	UARTPeriphID3	UART Peripheral Identification 3	
7F8h	UARTPCellID0	UART PrimeCell Identification 0	
7FAh	UARTPCellID1	UART PrimeCell Identification 1	
7FCh	UARTPCellID2	UART PrimeCell Identification 2	
7FEh	UARTPCellID3	UART PrimeCell Identification 3	

Complex bit access types are encoded to fit into small table cells. Table 18-3 shows the codes that are used for access types in this section.

Table 18-3. UART_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		

Table 18-3. UART_REGS Access Type Codes (continued)

Access Type	Code	Description
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

18.5.2.1 UARTDR Register (Offset = 0h) [Reset = 00000000h]

UARTDR is shown in Figure 18-4 and described in Table 18-4.

Return to the Summary Table.

IMPORTANT: This register is read sensitive. This register is the data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the UART.
For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

Figure 18-4. UARTDR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				OE	BE	PE	FE	DATA							
R-0h				R-0h	R-0h	R-0h	R-0h	R/W-0h							

Table 18-4. UARTDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11	OE	R	0h	UART Overrun Error 0 No data has been lost due to a FIFO overrun. 1 New data was received when the FIFO was full, resulting in data loss. Reset type: PER.RESET
10	BE	R	0h	UART Break Error 0 No break condition has occurred 1 A break condition has been detected, indicating that the receive data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits). In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the received data input goes to a 1 (marking state), and the next valid start bit is received. Reset type: PER.RESET
9	PE	R	0h	UART Parity Error 0 No parity error has occurred 1 The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register. In FIFO mode, this error is associated with the character at the top of the FIFO. Reset type: PER.RESET
8	FE	R	0h	UART Framing Error 0 No framing error has occurred 1 The received character does not have a valid stop bit (a valid stop bit is 1). Reset type: PER.RESET

Table 18-4. UARTDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	DATA	R/W	0h	Data Transmitted or Received Data that is to be transmitted via the UART is written to this field. When read, this field contains the data that was received by the UART. For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the UART. For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register. Reset type: PER.RESET

18.5.2.2 UARTRSR Register (Offset = 2h) [Reset = 00000000h]

UARTRSR is shown in Figure 18-5 and described in Table 18-5.

Return to the Summary Table.

The UARTRSR/UARTECR register is the receive status register/error clear register.
In addition to the UARTDR register, receive status can also be read from the UARTRSR register.
If the status is read from this register, then the status information corresponds to the entry read from UARTDR prior to reading UARTRSR. The status information for overrun is set immediately when an overrun condition occurs.
The UARTRSR register cannot be written.
A write of any value to the UARTECR register clears the framing, parity, break, and overrun errors.
All the bits are cleared on reset.

Figure 18-5. UARTRSR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												OE	BE	PE	FE
R-0h												R-0h	R-0h	R-0h	R-0h

Table 18-5. UARTRSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3	OE	R	0h	UART Overrun Error 0 No data has been lost due to a FIFO overrun. 1 New data was received when the FIFO was full, resulting in data loss. This bit is cleared by a write to UARTECR.The FIFO contents remain valid because no further data is written when the FIFO is full, only the contents of the shift register are overwritten. The CPU must read the data in order to empty the FIFO. Reset type: PER.RESET
2	BE	R	0h	UART Break Error 0 No break condition has occurred 1 A break condition has been detected, indicating that the receive data input was held Low for longer than a full-word transmission time (defined as start, data, parity, and stop bits). This bit is cleared to 0 by a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received. Reset type: PER.RESET
1	PE	R	0h	UART Parity Error 0 No parity error has occurred 1 The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register. This bit is cleared to 0 by a write to UARTECR. Reset type: PER.RESET

Table 18-5. UARTRSR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	FE	R	0h	UART Framing Error 0 No framing error has occurred 1 The received character does not have a valid stop bit (a valid stop bit is 1). This bit is cleared to 0 by a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO. Reset type: PER.RESET

18.5.2.3 UARTFR Register (Offset = Ch) [Reset = 00000090h]

UARTFR is shown in Figure 18-6 and described in Table 18-6.

Return to the Summary Table.

The UARTFR register is the flag register. After reset, the TXFF, RXFF, and BUSY bits are 0, and TXFE and RXFE bits are 1.

Figure 18-6. UARTFR Register

31	30	29	28	27	26	25	24
RESERVED			TXFIFOLVL				
R-0h			R-0h				
23	22	21	20	19	18	17	16
RESERVED			RXFIFOLVL				
R-0h			R-0h				
15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R-0h
7	6	5	4	3	2	1	0
TXFE	RXFF	TXFF	RXFE	BUSY	RESERVED	RESERVED	RESERVED
R-1h	R-0h	R-0h	R-1h	R-0h	R-0h	R-0h	R-0h

Table 18-6. UARTFR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	Reserved
28-24	TXFIFOLVL	R	0h	UART Transmit FIFO Level Indicates number of untransmitted TX FIFO entries (intended for CCS debugs) Reset type: PER.RESET
23-21	RESERVED	R	0h	Reserved
20-16	RXFIFOLVL	R	0h	UART Receive FIFO Level Indicates number of unread RX FIFO entries (intended for CCS debugs) Reset type: PER.RESET
15-9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	TXFE	R	1h	UART Transmit FIFO Empty The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register. 0 The transmitter has data to transmit. 1 If the FIFO is disabled (FEN is 0), the transmit holding register is empty. If the FIFO is enabled (FEN is 1), the transmit FIFO is empty. Reset type: PER.RESET
6	RXFF	R	0h	UART Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register. 0 The receiver can receive data. 1 If the FIFO is disabled (FEN is 0), the receive holding register is full. If the FIFO is enabled (FEN is 1), the receive FIFO is full. Reset type: PER.RESET

Table 18-6. UARTFR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	TXFF	R	0h	UART Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register. 0 The transmitter is not full. 1 If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full. Reset type: PER.RESET
4	RXFE	R	1h	UART Receive FIFO Empty The meaning of this bit depends on the state of the FEN bit in the UARTLCRH register. 0 The receiver is not empty. 1 If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty. Reset type: PER.RESET
3	BUSY	R	0h	UART Busy 0 The UART is not busy. 1 The UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent from the shift register. This bit is set as soon as the transmit FIFO becomes non-empty (regardless of whether UART is enabled). Reset type: PER.RESET
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

18.5.2.4 UARTILPR Register (Offset = 10h) [Reset = 00000000h]

UARTILPR is shown in Figure 18-7 and described in Table 18-7.

Return to the Summary Table.

The UARTILPR register stores the 8-bit low-power counter divisor value used to derive the low-power SIR pulse width clock.

Figure 18-7. UARTILPR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ILPDVSR							
R-0h																								R/W-0h							

Table 18-7. UARTILPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	ILPDVSR	R/W	0h	IrDA Low-Power Divisor This field contains the 8-bit low-power divisor value. The UARTILPR register stores the 8-bit low-power counter divisor value used to derive the low-power SIR pulse width clock by dividing down the system clock (SysClk). All the bits are cleared when reset. The internal IrLPBaud16 clock is generated by dividing down SysClk according to the low-power divisor value written to UARTILPR. The duration of SIR pulses generated when low-power mode is enabled is three times the period of the IrLPBaud16 clock. The low-power divisor value is calculated as follows: ILPDVSR = SysClk / FIrLPBaud16 where FIrLPBaud16 is nominally 1.8432 MHz. Because the IrLPBaud16 clock is used to sample transmitted data irrespective of mode, the ILPDVSR field must be programmed in both low power and normal mode,such that FIrLPBaud16 is between 1.42 and 2.12 MHz, resulting in a low-power pulse duration of 1.41-2.11 us (three times the period of IrLPBaud16). The minimum frequency of IrLPBaud16 ensures that pulses less than one period of IrLPBaud16 are rejected, but pulses greater than 1.4 us are accepted as valid pulses. Note: Zero is an illegal value. Programming a zero value results in no IrLPBaud16 pulses being generated Reset type: PER.RESET

18.5.2.5 UARTIBRD Register (Offset = 12h) [Reset = 00000000h]

UARTIBRD is shown in Figure 18-8 and described in Table 18-8.

Return to the Summary Table.

The UARTIBRD register is the integer part of the baud-rate divisor value. All the bits are cleared on reset. The minimum possible divide ratio is 1 (when UARTIBRD=0), in which case the UARTFBRD register is ignored. When changing the UARTIBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register.

Figure 18-8. UARTIBRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DIVINT															
R-0h																R/W-0h															

Table 18-8. UARTIBRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	DIVINT	R/W	0h	Integer Baud-Rate Divisor The minimum possible divide ratio is 1 (when UARTIBRD=0), in which case the UARTFBRD register is ignored. When changing the UARTIBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. Reset type: PER.RESET

18.5.2.6 UARTFBRD Register (Offset = 14h) [Reset = 00000000h]

UARTFBRD is shown in Figure 18-9 and described in Table 18-9.

Return to the Summary Table.

The UARTFBRD register is the fractional part of the baud-rate divisor value. All the bits are cleared on reset. When changing the UARTFBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. See 'Baud-Rate Generation' on page 1165 for configuration details.

Figure 18-9. UARTFBRD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										DIVFRAC					
R-0h										R/W-0h					

Table 18-9. UARTFBRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	DIVFRAC	R/W	0h	Fractional Baud-Rate Divisor The UARTFBRD register is the fractional part of the baud-rate divisor value. All the bits are cleared on reset. When changing the UARTFBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. Reset type: PER.RESET

18.5.2.7 UARTLCRH Register (Offset = 16h) [Reset = 00000000h]

UARTLCRH is shown in Figure 18-10 and described in Table 18-10.

Return to the Summary Table.

The UARTLCRH register is the line control register. Serial parameters such as data length, parity, and stop bit selection are implemented in this register.
When updating the baud-rate divisor (UARTIBRD and/or UARTIFRD), the UARTLCRH register must also be written. The write strobe for the baud-rate divisor registers is tied to the UARTLCRH register.

Figure 18-10. UARTLCRH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SPS	WLEN		FEN	STP2	EPS	PEN	BRK
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-10. UARTLCRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	SPS	R/W	0h	UART Stick Parity Select UART Stick Parity Select 0 Stick parity is disabled (default) 1 Stick parity is enabled. When bits 1, 2, and 7 of UARTLCRH are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set and 2 is cleared, the parity bit is transmitted and checked as a 1. Reset type: PER.RESET
6-5	WLEN	R/W	0h	UART Word Length The bits indicate the number of data bits transmitted or received in a frame as follows: 0x0 5 bits (default) 0x1 6 bits 0x2 7 bits 0x3 8 bits Reset type: PER.RESET
4	FEN	R/W	0h	UART Enable FIFOs 0 The FIFOs are disabled. The FIFOs become 1-byte-deep holding registers. 1 The transmit and receive FIFO buffers are enabled (FIFO mode). Reset type: PER.RESET
3	STP2	R/W	0h	UART Two Stop Bits Select 0 One stop bit is transmitted at the end of a frame. 1 Two stop bits are transmitted at the end of a frame. The receive logic does not check for two stop bits being received. Reset type: PER.RESET

Table 18-10. UARTLCRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	EPS	R/W	0h	UART Even Parity Select 0 Odd parity is performed, which checks for an odd number of 1s. 1 Even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. This bit has no effect when parity is disabled by the PEN bit. Reset type: PER.RESET
1	PEN	R/W	0h	UART Parity Enable 0 Parity is disabled and no parity bit is added to the data frame. 1 Parity checking and generation is enabled. Reset type: PER.RESET
0	BRK	R/W	0h	UART Send Break 0 Normal use. 1 A Low level is continually output on the UnTx signal, after completing transmission of the current character. For the proper execution of the break command, software must set this bit for at least two frames (character periods). Reset type: PER.RESET

18.5.2.8 UARTCTL Register (Offset = 18h) [Reset = 00000300h]

UARTCTL is shown in Figure 18-11 and described in Table 18-11.

Return to the Summary Table.

The UARTCTL register is the control register. All the bits are cleared on reset except for the Transmit Enable (TXE) and Receive Enable (RXE) bits, which are set.
To enable the UART module, the UARTEN bit must be set. If software requires a configuration change in the module, the UARTEN bit must be cleared before the configuration changes are written. If the UART is disabled during a transmit or receive operation, the current transaction is completed prior to the UART stopping.

Figure 18-11. UARTCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED		RESERVED	RESERVED	RXE	TXE
R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
LBE	RESERVED	HSE	EOT	RESERVED	SIRLP	SIREN	UARTEN
R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-11. UARTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13-12	RESERVED	R	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RXE	R/W	1h	UART Receive Enable 0 The receive section of the UART is disabled. 1 The receive section of the UART is enabled. If the UART is disabled in the middle of a receive, it completes the current character before stopping.To enable reception, the UARTEN bit must also be set. Reset type: PER.RESET
8	TXE	R/W	1h	UART Transmit Enable 0 The transmit section of the UART is disabled. 1 The transmit section of the UART is enabled. If the UART is disabled in the middle of a transmission, it completes the current character before stopping. To enable transmission, the UARTEN bit must also be set. Reset type: PER.RESET
7	LBE	R/W	0h	UART Loop Back Enable 0 Normal operation. 1 The UnTx path is fed through the UnRx path. Reset type: PER.RESET
6	RESERVED	R	0h	Reserved

Table 18-11. UARTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	HSE	R/W	0h	High-Speed Enable 0 The UART is clocked using the system clock divided by 16. 1 The UART is clocked using the system clock divided by 8. Reset type: PER.RESET
4	EOT	R/W	0h	End of Transmission This bit determines the behavior of the TXRIS bit in the UARTRIS register. 0 The TXRIS bit is set when the transmit FIFO condition specified in UARTIFLS is met. 1 The TXRIS bit is set only after all transmitted data, including stop bits, have cleared the serializer. Reset type: PER.RESET
3	RESERVED	R/W	0h	Reserved
2	SIRLP	R/W	0h	UART SIR Low-Power Mode This bit selects the IrDA encoding mode. 0 Low-level bits are transmitted as an active High pulse with a width of 3/16th of the bit period. 1 The UART operates in SIR Low-Power mode. Low-level bits are transmitted with a pulse width which is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate. Setting this bit uses less power, but might reduce transmission distances. Reset type: PER.RESET
1	SIREN	R/W	0h	UART SIR Enable 0 Normal operation. 1 The IrDA SIR block is enabled, and the UART will transmit and receive data using SIR protocol. Reset type: PER.RESET
0	UARTEN	R/W	0h	UART Enable 0 The UART is disabled. 1 The UART is enabled. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping. Reset type: PER.RESET

18.5.2.9 UARTIFLS Register (Offset = 1Ah) [Reset = 00000012h]

UARTIFLS is shown in Figure 18-12 and described in Table 18-12.

Return to the Summary Table.

The UARTIFLS register is the interrupt FIFO level select register. You can use this register to define the FIFO level at which the TXRIS and RXRIS bits in the UARTRIS register are triggered.
The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level.
For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered as the module is receiving the 9th character.
Therefore, changing the trigger level does not trigger an interrupt using the new level until another character is received.
Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

Figure 18-12. UARTIFLS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RXIFLSEL_FINE			
R-0h				R/W-0h			
15	14	13	12	11	10	9	8
RESERVED				TXIFLSEL_FINE			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED		RXIFLSEL				TXIFLSEL	
R-0h		R/W-2h				R/W-2h	

Table 18-12. UARTIFLS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-16	RXIFLSEL_FINE	R/W	0h	Value Description 0x0 RX FIFO FULL - 16 filled spots 0x1 RX FIFO greater than or equal to 1/16 full - at least 1 filled spots 0x2 RX FIFO greater than or equal to 2/16 full - at least 2 filled spots ... 0xE RX FIFO greater than or equal to 14/16 full - at least 14 filled spots 0xF RX FIFO greater than or equal to 15/16 full - at least 15 filled spots Note : This field is considered only if RXIFLSEL = 0x5 Reset type: PER.RESET
15-12	RESERVED	R	0h	Reserved
11-8	TXIFLSEL_FINE	R/W	0h	Value Description 0x0 TX FIFO empty - 16 empty slots 0x1 TX FIFO less than or equal to 1/16 full - atleast 15 empty slots 0x2 TX FIFO less than or equal to 2/16 full - atleast 14 empty slots ... 0xE TX FIFO less than or equal to 14/16 full - atleast 2 empty slots 0xF TX FIFO less than or equal to 15/16 full - atleast 1 empty slots Note : This field is considered only if TXIFLSEL = 0x5 Reset type: PER.RESET
7-6	RESERVED	R	0h	Reserved

Table 18-12. UARTIFLS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-3	RXIFLSEL	R/W	2h	The trigger points for the receive interrupt are as follows: Value Description 0x0 RX FIFO greater than or equal to 1/8 full - at least 2 filled spots 0x1 RX FIFO greater than or equal to 1/4 full - at least 4 filled spots 0x2 RX FIFO greater than or equal to 1/2 full - at least 8 filled spots (default) 0x3 RX FIFO greater than or equal to 3/4 full - at least 12 filled spots 0x4 RX FIFO greater than or equal to 7/8 full - at least 14 filled spots 0x5 Fine-grained RX FIFO Mode (Refer to RXIFLSEL_FINE field of this register) 0x6-0x7 Reserved Reset type: PER.RESET
2-0	TXIFLSEL	R/W	2h	The trigger points for the transmit interrupt are as follows: Value Description 0x0 TX FIFO less than or equal to 1/8 full - at least 14 empty spots 0x1 TX FIFO less than or equal to 1/4 full - at least 12 empty spots 0x2 TX FIFO less than or equal to 1/2 full - at least 8 empty spots (default) 0x3 TX FIFO less than or equal to 3/4 full - at least 4 empty spots 0x4 TX FIFO less than or equal to 7/8 full - at least 2 empty spots 0x5 Fine-grained TX FIFO Mode (Refer to TXIFLSEL_FINE field of this register) 0x6-0x7 Reserved Note: If the EOT bit in UARTCTL is set, the transmit interrupt is generated once the FIFO is completely empty and all data including stop bits have left the transmit serializer. In this case, the setting of TXIFLSEL is ignored. Reset type: PER.RESET

18.5.2.10 UARTIM Register (Offset = 1Ch) [Reset = 00000000h]

UARTIM is shown in Figure 18-13 and described in Table 18-13.

Return to the Summary Table.

The UARTIM register is the interrupt mask set/clear register.
On a read, this register gives the current value of the mask on the relevant interrupt. Setting a bit allows the corresponding raw interrupt signal to be routed to the interrupt controller. Clearing a bit prevents the raw interrupt signal from being sent to the interrupt controller.

Figure 18-13. UARTIM Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						DMATXIM	DMARXIM
R-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED			9BITIM	RESERVED	OEIM	BEIM	PEIM
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
FEIM	RTIM	TXIM	RXIM	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 18-13. UARTIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	DMATXIM	R/W	0h	Transmit DMA Interrupt Mask 0 The DMATXRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the DMATXRIS bit in the UARTRIS register is set. Reset type: PER.RESET
16	DMARXIM	R/W	0h	Receive DMA Interrupt Mask 0 The DMARXRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the DMARXRIS bit in the UARTRIS register is set. Reset type: PER.RESET
15-13	RESERVED	R	0h	Reserved
12	9BITIM	R/W	0h	9-Bit Mode Interrupt Mask 0 The 9BITRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the 9BITRIS bit in the UARTRIS register is set. Reset type: PER.RESET
11	RESERVED	R/W	0h	Reserved
10	OEIM	R/W	0h	UART Overrun Error Interrupt Mask 0 The OERIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the OERIS bit in the UARTRIS register is set. Reset type: PER.RESET

Table 18-13. UARTIM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	BEIM	R/W	0h	UART Break Error Interrupt Mask 0 The BERIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the BERIS bit in the UARTRIS register is set. Reset type: PER.RESET
8	PEIM	R/W	0h	UART Parity Error Interrupt Mask 0 The PERIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the PERIS bit in the UARTRIS register is set. Reset type: PER.RESET
7	FEIM	R/W	0h	UART Framing Error Interrupt Mask 0 The FERIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the FERIS bit in the UARTRIS register is set. Reset type: PER.RESET
6	RTIM	R/W	0h	UART Receive Time-Out Interrupt Mask 0 The RTRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the RTRIS bit in the UARTRIS register is set. Reset type: PER.RESET
5	TXIM	R/W	0h	UART Transmit Interrupt Mask 0 The TXRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the TXRIS bit in the UARTRIS register is set. Reset type: PER.RESET
4	RXIM	R/W	0h	UART Receive Interrupt Mask 0 The RXRIS interrupt is suppressed and not sent to the interrupt controller. 1 An interrupt is sent to the interrupt controller when the RXRIS bit in the UARTRIS register is set. Reset type: PER.RESET
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

18.5.2.11 UARTRIS Register (Offset = 1Eh) [Reset = 00000000h]

UARTRIS is shown in Figure 18-14 and described in Table 18-14.

Return to the Summary Table.

The UARTRIS register is the raw interrupt status register. On a read, this register gives the current raw status value of the corresponding interrupt. A write has no effect.

Figure 18-14. UARTRIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						DMATXRIS	DMARXRIS
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			9BITRIS	RESERVED	OERIS	BERIS	PERIS
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FERIS	RTRIS	TXRIS	RXRIS	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 18-14. UARTRIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	DMATXRIS	R	0h	Transmit DMA Raw Interrupt Status 0 No interrupt 1 The transmit DMA has completed. This bit is cleared by writing a 1 to the DMATXIC bit in the UARTICR register. Reset type: PER.RESET
16	DMARXRIS	R	0h	Receive DMA Raw Interrupt Status 0 No interrupt 1 The receive DMA has completed. This bit is cleared by writing a 1 to the DMARXIC bit in the UARTICR register. Reset type: PER.RESET
15-13	RESERVED	R	0h	Reserved
12	9BITRIS	R	0h	9-Bit Mode Raw Interrupt Status 0 No interrupt 1 A receive address match has occurred. This bit is cleared by writing a 1 to the 9BITIC bit in the UARTICR register. Reset type: PER.RESET
11	RESERVED	R	0h	Reserved
10	OERIS	R	0h	UART Overrun Error Raw Interrupt Status 0 No interrupt 1 An overrun error has occurred. This bit is cleared by writing a 1 to the OEIC bit in the UARTICR register. Reset type: PER.RESET

Table 18-14. UARTRIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	BERIS	R	0h	UART Break Error Raw Interrupt Status 0 No interrupt 1 A break error has occurred. This bit is cleared by writing a 1 to the BEIC bit in the UARTICR register. Reset type: PER.RESET
8	PERIS	R	0h	UART Parity Error Raw Interrupt Status 0 No interrupt 1 A parity error has occurred. This bit is cleared by writing a 1 to the PEIC bit in the UARTICR register. Reset type: PER.RESET
7	FERIS	R	0h	UART Framing Error Raw Interrupt Status 0 No interrupt 1 A framing error has occurred. This bit is cleared by writing a 1 to the FEIC bit in the UARTICR register. Reset type: PER.RESET
6	RTRIS	R	0h	UART Receive Time-Out Raw Interrupt Status 0 No interrupt 1 A receive time out has occurred. This bit is cleared by writing a 1 to the RTIC bit in the UARTICR register. Reset type: PER.RESET
5	TXRIS	R	0h	UART Transmit Raw Interrupt Status 0 No interrupt 1 If the EOT bit in the UARTCTL register is clear, the transmit FIFO level has passed through the condition defined in the UARTIFLS register.If the EOT bit is set, the last bit of all transmitted data and flags has left the serializer. This bit is cleared by writing a 1 to the TXIC bit in the UARTICR register or by writing data to the transmit FIFO until it becomes greater than the trigger level, if the FIFO is enabled, or by writing a single byte if the FIFO is disabled. Reset type: PER.RESET
4	RXRIS	R	0h	UART Receive Raw Interrupt Status 0 No interrupt 1 The receive FIFO level has passed through the condition defined in the UARTIFLS register. This bit is cleared by writing a 1 to the RXIC bit in the UARTICR register or by reading data from the receive FIFO until it becomes less than the trigger level, if the FIFO is enabled, or by reading a single byte if the FIFO is disabled. Reset type: PER.RESET
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

18.5.2.12 UARTMIS Register (Offset = 20h) [Reset = 00000000h]

UARTMIS is shown in Figure 18-15 and described in Table 18-15.

Return to the Summary Table.

The UARTMIS register is the masked interrupt status register. On a read, this register gives the current masked status value of the corresponding interrupt. A write has no effect.

Figure 18-15. UARTMIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						DMATXMIS	DMARXMIS
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			9BITMIS	RESERVED	OEMIS	BEMIS	PEMIS
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FEMIS	RTMIS	TXMIS	RXMIS	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 18-15. UARTMIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	DMATXMIS	R	0h	Transmit DMA Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to the completion of the transmit DMA. This bit is cleared by writing a 1 to the DMATXIC bit in the UARTICR register. Reset type: PER.RESET
16	DMARXMIS	R	0h	Receive DMA Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to the completion of the receive DMA. This bit is cleared by writing a 1 to the DMARXIC bit in the UARTICR register. Reset type: PER.RESET
15-13	RESERVED	R	0h	Reserved
12	9BITMIS	R	0h	9-Bit Mode Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a receive address match. This bit is cleared by writing a 1 to the 9BITIC bit in the UARTICR register. Reset type: PER.RESET
11	RESERVED	R	0h	Reserved
10	OEMIS	R	0h	UART Overrun Error Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to an overrun error. This bit is cleared by writing a 1 to the OEIC bit in the UARTICR register. Reset type: PER.RESET

Table 18-15. UARTMIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	BEMIS	R	0h	UART Break Error Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a break error. This bit is cleared by writing a 1 to the BEIC bit in the UARTICR register. Reset type: PER.RESET
8	PEMIS	R	0h	UART Parity Error Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a parity error. This bit is cleared by writing a 1 to the PEIC bit in the UARTICR register. Reset type: PER.RESET
7	FEMIS	R	0h	UART Framing Error Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a framing error. This bit is cleared by writing a 1 to the FEIC bit in the UARTICR register. Reset type: PER.RESET
6	RTMIS	R	0h	UART Receive Time-Out Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to a receive time out. This bit is cleared by writing a 1 to the RTIC bit in the UARTICR register. Reset type: PER.RESET
5	TXMIS	R	0h	UART Transmit Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to passing through the specified transmit FIFO level (if the EOT bit is clear) or due to the transmission of the last data bit (if the EOT bit is set). This bit is cleared by writing a 1 to the TXIC bit in the UARTICR register or by writing data to the transmit FIFO until it becomes greater than the trigger level, if the FIFO is enabled, or by writing a single byte if the FIFO is disabled. Reset type: PER.RESET
4	RXMIS	R	0h	UART Receive Masked Interrupt Status 0 An interrupt has not occurred or is masked. 1 An unmasked interrupt was signaled due to passing through the specified receive FIFO level. This bit is cleared by writing a 1 to the RXIC bit in the UARTICR register or by reading data from the receive FIFO until it becomes less than the trigger level, if the FIFO is enabled, or by reading a single byte if the FIFO is disabled. Reset type: PER.RESET
3	RESERVED	R	0h	Reserved
2	RESERVED	R	0h	Reserved
1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

18.5.2.13 UARTICR Register (Offset = 22h) [Reset = 00000000h]

UARTICR is shown in Figure 18-16 and described in Table 18-16.

Return to the Summary Table.

The UARTICR register is the interrupt clear register. On a write of 1, the corresponding interrupt (both raw interrupt and masked interrupt, if enabled) is cleared. A write of 0 has no effect.

Figure 18-16. UARTICR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						DMATXIC	DMARXIC
R-0h						R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED			9BITIC	EOTIC	OEIC	BEIC	PEIC
R-0h			R/W-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
FEIC	RTIC	TXIC	RXIC	RESERVED	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 18-16. UARTICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17	DMATXIC	R-0/W1S	0h	Transmit DMA Interrupt Clear Writing a 1 to this bit clears the DMATXRIS bit in the UARTRIS register and the DMATXMIS bit in the UARTMIS register. Reset type: PER.RESET
16	DMARXIC	R-0/W1S	0h	Receive DMA Interrupt Clear Writing a 1 to this bit clears the DMARXRIS bit in the UARTRIS register and the DMARXMIS bit in the UARTMIS register. Reset type: PER.RESET
15-13	RESERVED	R	0h	Reserved
12	9BITIC	R/W	0h	9-Bit Mode Interrupt Clear Writing a 1 to this bit clears the 9BITRIS bit in the UARTRIS register and the 9BITMIS bit in the UARTMIS register. Reset type: PER.RESET
11	EOTIC	R-0/W1S	0h	End of Transmission Interrupt Clear Writing a 1 to this bit clears the EOTRIS bit in the UARTRIS register and the EOTMIS bit in the UARTMIS register. Reset type: PER.RESET
10	OEIC	R-0/W1S	0h	Overrun Error Interrupt Clear Writing a 1 to this bit clears the OERIS bit in the UARTRIS register and the OEMIS bit in the UARTMIS register. Reset type: PER.RESET
9	BEIC	R-0/W1S	0h	Break Error Interrupt Clear Writing a 1 to this bit clears the BERIS bit in the UARTRIS register and the BEMIS bit in the UARTMIS register. Reset type: PER.RESET
8	PEIC	R-0/W1S	0h	Parity Error Interrupt Clear Writing a 1 to this bit clears the PERIS bit in the UARTRIS register and the PEMIS bit in the UARTMIS register. Reset type: PER.RESET

Table 18-16. UARTICR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	FEIC	R-0/W1S	0h	Framing Error Interrupt Clear Writing a 1 to this bit clears the FERIS bit in the UARTRIS register and the FEMIS bit in the UARTMIS register. Reset type: PER.RESET
6	RTIC	R-0/W1S	0h	Receive Time-Out Interrupt Clear Writing a 1 to this bit clears the RTRIS bit in the UARTRIS register and the RTMIS bit in the UARTMIS register. Reset type: PER.RESET
5	TXIC	R-0/W1S	0h	Transmit Interrupt Clear Writing a 1 to this bit clears the TXRIS bit in the UARTRIS register and the TXMIS bit in the UARTMIS register. Reset type: PER.RESET
4	RXIC	R-0/W1S	0h	Receive Interrupt Clear Writing a 1 to this bit clears the RXRIS bit in the UARTRIS register and the RXMIS bit in the UARTMIS register. Reset type: PER.RESET
3	RESERVED	R-0/W1S	0h	Reserved
2	RESERVED	R-0/W1S	0h	Reserved
1	RESERVED	R-0/W1S	0h	Reserved
0	RESERVED	R-0/W1S	0h	Reserved

18.5.2.14 UARTDMACTL Register (Offset = 24h) [Reset = 00000000h]

UARTDMACTL is shown in Figure 18-17 and described in Table 18-17.

Return to the Summary Table.

UART DMA Control

Figure 18-17. UARTDMACTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					DMAERR	TXDMAE	RXDMAE
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 18-17. UARTDMACTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	DMAERR	R/W	0h	DMA on Error 0 DMA receive requests are unaffected when a receive error occurs. 1 DMA receive requests are automatically disabled when a receive error occurs. Reset type: PER.RESET
1	TXDMAE	R/W	0h	Transmit DMA Enable 0 DMA for the transmit FIFO is disabled. 1 DMA for the transmit FIFO is enabled. Reset type: PER.RESET
0	RXDMAE	R/W	0h	Receive DMA Enable 0 DMA for the receive FIFO is disabled. 1 DMA for the receive FIFO is enabled. Reset type: PER.RESET

18.5.2.15 UART_GLB_INT_EN Register (Offset = 40h) [Reset = 00000000h]

UART_GLB_INT_EN is shown in Figure 18-18 and described in Table 18-18.

Return to the Summary Table.

The UART_GLB_INT_EN register is used to enable interrupt from UART to PIE

Figure 18-18. UART_GLB_INT_EN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							RESERVED
R-0h							R/W-0h

Table 18-18. UART_GLB_INT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	RESERVED	R/W	0h	Reserved

18.5.2.16 UART_GLB_INT_FLG Register (Offset = 42h) [Reset = 00000000h]

UART_GLB_INT_FLG is shown in Figure 18-19 and described in Table 18-19.

Return to the Summary Table.

The UART_GLB_INT_FLG register contains the current status of the UART interrupt

Figure 18-19. UART_GLB_INT_FLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							INT_FLG
R-0h							R-0h

Table 18-19. UART_GLB_INT_FLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	INT_FLG	R	0h	Global Interrupt Flag for UART INT. This bit determines whether the SINTREQUEST is generated by UART This bit can be cleared by writing a 1 to the corresponding bit in the UART_GLB_INT_CLR register. Reset type: SYSRSn

18.5.2.17 UART_GLB_INT_CLR Register (Offset = 44h) [Reset = 00000000h]

UART_GLB_INT_CLR is shown in Figure 18-20 and described in Table 18-20.

Return to the Summary Table.

The UART_GLB_INT_CLR register is used to clear the interrupt flags in UART_GLB_INT_FLG register.

Figure 18-20. UART_GLB_INT_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							INT_FLG_CLR
R-0h							R/W1C-0h

Table 18-20. UART_GLB_INT_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	INT_FLG_CLR	R/W1C	0h	Global Interrupt flag clear for UART INT. This bit is used to clear the corresponding bit in the UART_GLB_INT_FLG register. Write 1 to clear the INT_FLG bit. Writing 0 has no effect. Reset type: SYSRSn

18.5.2.18 UART9BITADDR Register (Offset = 52h) [Reset = 00000000h]

UART9BITADDR is shown in Figure 18-21 and described in Table 18-21.

Return to the Summary Table.

The UART9BITADDR register is used to write the specific address that should be matched with the receiving byte when the 9-bit Address Mask (UART9BITAMASK) is set to 0xFF. This register is used in conjunction with UART9BITAMASK to form a match for address-byte received.

Figure 18-21. UART9BITADDR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
9BITEN	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
ADDR							
R/W-0h							

Table 18-21. UART9BITADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	9BITEN	R/W	0h	Enable 9-Bit Mode 0 9-bit mode is disabled. 1 9-bit mode is enabled. Reset type: PER.RESET
14-8	RESERVED	R	0h	Reserved
7-0	ADDR	R/W	0h	Self Address for 9-Bit Mode This field contains the address that should be matched when UART9BITAMASK is 0xFF. Reset type: PER.RESET

18.5.2.19 UART9BITAMASK Register (Offset = 54h) [Reset = 000000FFh]

UART9BITAMASK is shown in Figure 18-22 and described in Table 18-22.

Return to the Summary Table.

The UART9BITAMASK register is used to enable the address mask for 9-bit mode. The address bits are masked to create a set of addresses to be matched with the received address byte.

Figure 18-22. UART9BITAMASK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MASK							
R-0h																								R/W-FFh							

Table 18-22. UART9BITAMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	MASK	R/W	FFh	Self Address Mask for 9-Bit Mode This field contains the address mask that creates a set of addresses that should be matched. Reset type: PER.RESET

18.5.2.20 UARTPP Register (Offset = 7E0h) [Reset = 00000002h]

UARTPP is shown in Figure 18-23 and described in Table 18-23.

Return to the Summary Table.

The UARTPP register provides information regarding the properties of the UART module.

Figure 18-23. UARTPP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												MSE	MS	NB	SC
R-0h												R-0h	R-0h	R-1h	R-0h

Table 18-23. UARTPP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3	MSE	R	0h	Modem Support Extended 0 The UART module does not provide extended support for modem control. 1 The UART module provides extended support for modem control including UARTnDTR, UARTnDSR, UARTnDCD, and UARTnRI. Reset type: PER.RESET
2	MS	R	0h	Modem Support 0 The UART module does not provide support for modem control. 1 The UART module provides support for modem control including UARTnRTS and UARTnCTS. Reset type: PER.RESET
1	NB	R	1h	9-Bit Support 0 The UART module does not provide support for the transmission of 9-bit data for RS-485 support. 1 The UART module provides support for the transmission of 9-bit data for RS-485 support. Reset type: PER.RESET
0	SC	R	0h	Smart Card Support 0 The UART module does not provide smart card support. 1 The UART module provides smart card support. Reset type: PER.RESET

18.5.2.21 UARTPeriphID4 Register (Offset = 7E8h) [Reset = 00000060h]

UARTPeriphID4 is shown in Figure 18-24 and described in Table 18-24.

Return to the Summary Table.

UART Peripheral Identification 4

Figure 18-24. UARTPeriphID4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID4							
R-0h																								R-60h							

Table 18-24. UARTPeriphID4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID4	R	60h	UART Peripheral ID Register [7:0] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

18.5.2.22 UARTPeriphID5 Register (Offset = 7EAh) [Reset = 00000000h]

UARTPeriphID5 is shown in Figure 18-25 and described in Table 18-25.

Return to the Summary Table.

UART Peripheral Identification 5

Figure 18-25. UARTPeriphID5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID5							
R-0h																								R-0h							

Table 18-25. UARTPeriphID5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID5	R	0h	UART Peripheral ID Register [15:8] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

18.5.2.23 UARTPeriphID6 Register (Offset = 7ECh) [Reset = 00000000h]

UARTPeriphID6 is shown in Figure 18-26 and described in Table 18-26.

Return to the Summary Table.

UART Peripheral Identification 6

Figure 18-26. UARTPeriphID6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID6							
R-0h																								R-0h							

Table 18-26. UARTPeriphID6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID6	R	0h	UART Peripheral ID Register [23:16] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

18.5.2.24 UARTPeriphID7 Register (Offset = 7EEh) [Reset = 00000000h]

UARTPeriphID7 is shown in Figure 18-27 and described in Table 18-27.

Return to the Summary Table.

UART Peripheral Identification 7

Figure 18-27. UARTPeriphID7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID7							
R-0h																								R-0h							

Table 18-27. UARTPeriphID7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID7	R	0h	UART Peripheral ID Register [31:24] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

18.5.2.25 UARTPeriphID0 Register (Offset = 7F0h) [Reset = 00000011h]

UARTPeriphID0 is shown in Figure 18-28 and described in Table 18-28.

Return to the Summary Table.

UART Peripheral Identification 0

Figure 18-28. UARTPeriphID0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID0							
R-0h																								R-11h							

Table 18-28. UARTPeriphID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID0	R	11h	UART Peripheral ID Register [7:0] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

18.5.2.26 UARTPeriphID1 Register (Offset = 7F2h) [Reset = 00000000h]

UARTPeriphID1 is shown in Figure 18-29 and described in Table 18-29.

Return to the Summary Table.

UART Peripheral Identification 1

Figure 18-29. UARTPeriphID1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID1							
R-0h																								R-0h							

Table 18-29. UARTPeriphID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID1	R	0h	UART Peripheral ID Register [15:8] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

18.5.2.27 UARTPeriphID2 Register (Offset = 7F4h) [Reset = 00000018h]

UARTPeriphID2 is shown in Figure 18-30 and described in Table 18-30.

Return to the Summary Table.

UART Peripheral Identification 2

Figure 18-30. UARTPeriphID2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID2							
R-0h																								R-18h							

Table 18-30. UARTPeriphID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID2	R	18h	UART Peripheral ID Register [23:16] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

18.5.2.28 UARTPeriphID3 Register (Offset = 7F6h) [Reset = 00000001h]

UARTPeriphID3 is shown in Figure 18-31 and described in Table 18-31.

Return to the Summary Table.

UART Peripheral Identification 3

Figure 18-31. UARTPeriphID3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PID3							
R-0h																								R-1h							

Table 18-31. UARTPeriphID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	PID3	R	1h	UART Peripheral ID Register [31:24] Can be used by software to identify the presence of this peripheral. Reset type: PER.RESET

18.5.2.29 UARTPCellID0 Register (Offset = 7F8h) [Reset = 0000000Dh]

UARTPCellID0 is shown in Figure 18-32 and described in Table 18-32.

Return to the Summary Table.

UART PrimeCell Identification 0

Figure 18-32. UARTPCellID0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CID0							
R-0h																								R-Dh							

Table 18-32. UARTPCellID0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	CID0	R	Dh	UART PrimeCell ID Register [7:0] Provides software a standard cross-peripheral identification system. Reset type: PER.RESET

18.5.2.30 UARTPCellID1 Register (Offset = 7FAh) [Reset = 000000F0h]

UARTPCellID1 is shown in Figure 18-33 and described in Table 18-33.

Return to the Summary Table.

UART PrimeCell Identification 1

Figure 18-33. UARTPCellID1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CID1							
R-0h																								R-F0h							

Table 18-33. UARTPCellID1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	CID1	R	F0h	UART PrimeCell ID Register [15:8] Provides software a standard cross-peripheral identification system. Reset type: PER.RESET

18.5.2.31 UARTPCellID2 Register (Offset = 7FCh) [Reset = 00000005h]

UARTPCellID2 is shown in Figure 18-34 and described in Table 18-34.

Return to the Summary Table.

UART PrimeCell Identification 2

Figure 18-34. UARTPCellID2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CID2							
R-0h																								R-5h							

Table 18-34. UARTPCellID2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	CID2	R	5h	UART PrimeCell ID Register [23:16] Provides software a standard cross-peripheral identification system. Reset type: PER.RESET

18.5.2.32 UARTPCellID3 Register (Offset = 7FEh) [Reset = 000000B1h]

UARTPCellID3 is shown in Figure 18-35 and described in Table 18-35.

Return to the Summary Table.

UART PrimeCell Identification 3

Figure 18-35. UARTPCellID3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CID3							
R-0h																								R-B1h							

Table 18-35. UARTPCellID3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	CID3	R	B1h	UART PrimeCell ID Register [31:24] Provides software a standard cross-peripheral identification system. Reset type: PER.RESET

18.5.3 UART_REGS_WRITE Registers

Table 18-36 lists the memory-mapped registers for the UART_REGS_WRITE registers. All register offset addresses not listed in Table 18-36 should be considered as reserved locations and the register contents should not be modified.

Table 18-36. UART_REGS_WRITE Registers

Offset	Acronym	Register Name	Write Protection
2h	UARTECR	UART Error Clear	

Complex bit access types are encoded to fit into small table cells. Table 18-37 shows the codes that are used for access types in this section.

Table 18-37. UART_REGS_WRITE Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

18.5.3.1 UARTECR Register (Offset = 2h) [Reset = 00000000h]

UARTECR is shown in Figure 18-36 and described in Table 18-38.

Return to the Summary Table.

The UARTECR register is the error clear register.
In addition to the UARTDR register, receive status can also be read from the UARTRSR register. If the status is read from this register, then the status information corresponds to the entry read from UARTDR prior to reading UARTRSR. The status information for overrun is set immediately when an overrun condition occurs.
The UARTRSR register cannot be written.
A write of any value to the UARTECR register clears the framing, parity, break, and overrun errors.
All the bits are cleared on reset.

Figure 18-36. UARTECR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DATA							
R-0/W-0h																								R-0/W-0h							

Table 18-38. UARTECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0/W	0h	Reserved
7-0	DATA	R-0/W	0h	Error Clear A write to this register of any data clears the framing, parity, break, and overrun flags. Reset type: PER.RESET

Chapter 19

Serial Peripheral Interface (SPI)

This chapter describes the serial peripheral interface (SPI) which is a high-speed synchronous serial input and output (I/O) port that allows a serial bit stream of programmed length (one to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communications between the MCU controller and external peripherals or another controller. Typical applications include external I/O or peripheral expansion using devices such as shift registers, display drivers, and analog-to-digital converters (ADCs). Multi-device communications are supported by the controller or peripheral operation of the SPI. The port supports a -level, receive and transmit FIFO for reducing CPU servicing overhead.

19.1 Introduction

19.1.1 Features

The SPI module features include:

- SPIPOCI: SPI peripheral-output/controller-input pin
- SPIPICO: SPI peripheral-input/controller-output pin
- $\overline{\text{SPIPTE}}$: SPI peripheral transmit-enable pin
- SPICLK: SPI serial-clock pin

Note

All four pins can be used as GPIO if the SPI module is not used.

- Two operational modes: Controller and Peripheral
- Baud rate: 125 different programmable rates. The maximum baud rate that can be employed is limited by the maximum speed of the I/O buffers used on the SPI pins. See the device data sheet for more details.
- Data word length: 1 to 16 data bits
- Four clocking schemes (controlled by clock polarity and clock phase bits) include:
  - Falling edge without phase delay: SPICLK active-high. SPI transmits data on the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
  - Falling edge with phase delay: SPICLK active-high. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge without phase delay: SPICLK inactive-low. SPI transmits data on the rising edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
  - Rising edge with phase delay: SPICLK inactive-low. SPI transmits data one half-cycle ahead of the rising edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
- Simultaneous receive and transmit operation (transmit function can be disabled in software)
- Transmitter and receiver operations are accomplished through either interrupt- driven or polled algorithm
- 16-level transmit/receive FIFO
- DMA support
- High-speed mode
- Delayed transmit control
- $\overline{\text{SPIPTE}}$ inversion for digital audio interface receive mode on devices with two SPI modules

19.1.2 Block Diagram

Figure 19-1 shows the SPI CPU interfaces.

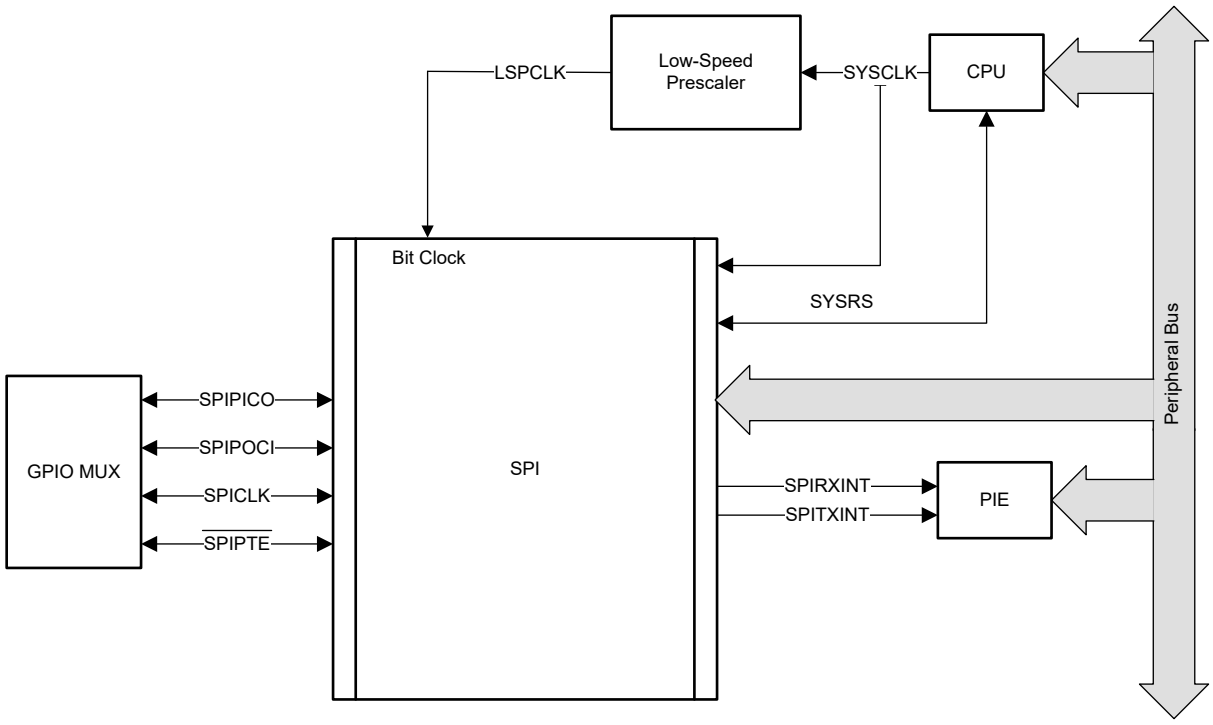


Figure 19-1. SPI CPU Interface

19.2 System-Level Integration

This section describes the various functionality that is applicable to the device integration. These features require configuration of other modules in the device that are not within the scope of this chapter.

19.2.1 SPI Module Signals

Table 19-1 classifies and provides a summary of the SPI module signals.

Table 19-1. SPI Module Signal Summary

Signal Name	Description
External Signals	
SPICLK	SPI clock
SPIPICO	SPI peripheral in, controller out
SPIPOCI	SPI peripheral out, controller in
$\overline{\text{SPIPTE}}$	SPI peripheral transmit enable
Control	
SPI Clock Rate	LSPCLK
Interrupt Signals	
SPIINT/SPIRXINT	Transmit interrupt/ Receive Interrupt in non FIFO mode (referred to as SPIINT)
	Receive interrupt in FIFO mode
SPITXINT	Transmit interrupt in FIFO mode
DMA Triggers	
SPITXDMA	Transmit request to DMA
SPIRXDMA	Receive request to DMA

Special Considerations

The $\overline{\text{SPIPTE}}$ signal provides the ability to gate any spurious clock and data pulses when the SPI is in peripheral mode. A HIGH logic signal on $\overline{\text{SPIPTE}}$ does not allow the peripheral to receive data. This prevents the SPI peripheral from losing synchronization with the controller. TI does not recommend that the $\overline{\text{SPIPTE}}$ always be tied to the active state.

If the SPI peripheral does ever lose synchronization with the controller, toggling SPISWRESET resets the internal bit counter as well as the various status flags in the module. By resetting the bit counter, the SPI interprets the next clock transition as the first bit of a new transmission. The register bit fields that are reset by SPISWRESET are found in Section 19.6.

Configuring a GPIO to Emulate $\overline{\text{SPIPTE}}$

In many systems, a SPI controller can be connected to multiple SPI peripherals using multiple instances of $\overline{\text{SPIPTE}}$. Though this SPI module does not natively support multiple $\overline{\text{SPIPTE}}$ signals, it is possible to emulate this behavior in software using GPIOs. In this configuration, the SPI must be configured as the controller. Rather than using the GPIO Mux to select $\overline{\text{SPIPTE}}$, the application can configure pins to be GPIO outputs, one GPIO per SPI peripheral. Before transmitting any data, the application can drive the desired GPIO to the active state. Immediately after the transmission has been completed, the GPIO chip select can be driven to the inactive state. This process can be repeated for many peripherals that share the SPICLK, SPIPICO, and SPIPOCI lines.

19.2.2 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

Some IO functionality is defined by GPIO register settings independent of this peripheral. For input signals, the GPIO input qualification must be set to asynchronous mode by setting the appropriate GPxQSELn register bits to 11b. The internal pullups can be configured in the GPyPUD register.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

19.2.2.1 GPIOs Required for High-Speed Mode

The high-speed mode of the SPI is available on all GPIO mux options. To enable the high-speed enhancements, set SPICCR.HS_MODE to 1. Make sure that the capacitive loading on the pin does not exceed the value stated in the device data sheet.

When not operating in high-speed mode or if the capacitive loading on the pins exceed the value stated in the device data sheet, SPICCR.HS_MODE can be set to 0.

19.2.3 SPI Interrupts

This section includes information on the available interrupts present in the SPI module. The SPI module contains two interrupt lines: SPIINT/SPIRXINT and SPITXINT. When the SPI is operating in non-FIFO mode, all available interrupts are routed together to generate the single SPIINT interrupt. When FIFO mode is used, both SPIRXINT and SPITXINT can be generated.

SPIINT/SPIRXINT

When the SPI is operating in non-FIFO mode, the interrupt generated is called SPIINT. If FIFO enhancements are enabled, the interrupt is called SPIRXINT. These interrupts share the same interrupt vector in the Peripheral Interrupt Expansion (PIE) block.

In non-FIFO mode, two conditions can trigger an interrupt: a transmission is complete (INT_FLAG), or there is overrun in the receiver (OVERRUN_FLAG). Both of these conditions share the same interrupt vector: SPIINT.

The transmission complete flag (INT_FLAG) indicates that the SPI has completed sending or receiving the last bit and is ready to be serviced. At the same time this bit is set, the received character is placed in the receiver buffer (SPIRXBUF). The INT_FLAG generates an interrupt on the SPIINT vector if the SPIINTENA bit is set.

The receiver overrun flag (OVERRUN_FLAG) indicates that a transmit or receive operation has completed before the previous character has been read from the buffer. The OVERRUN_FLAG generates an interrupt on the SPIINT vector if the OVERRUNINTENA bit is set and OVERRUN_FLAG was previously cleared.

In FIFO mode, the SPI can interrupt the CPU upon a match condition between the current receive FIFO status (RXFFST) and the receive FIFO interrupt level (RXFFIL). If RXFFST is greater than or equal to RXFFIL, the receive FIFO interrupt flag (RXFFINT) is set. SPIRXINT is triggered in the PIE block, if RXFFINT is set and the receive FIFO interrupt is enabled (RXFFIENA = 1).

SPITXINT

The SPITXINT interrupt is not available when the SPI is operating in non-FIFO mode.

In FIFO mode, the SPITXINT behavior is similar to the SPIRXINT. SPITXINT is generated upon a match condition between the current transmit FIFO status (TXFFST) and the transmit FIFO interrupt level (TXFFIL). If TXFFST is less than or equal to TXFFIL, the transmit FIFO interrupt flag (TXFFINT) is set. SPITXINT is triggered in the PIE block, if TXFFINT is set and the transmit FIFO interrupt is enabled in the SPI module (TXFFIENA = 1).

Figure 19-2 and Table 19-2 show how these control bits influence the SPI interrupt generation.

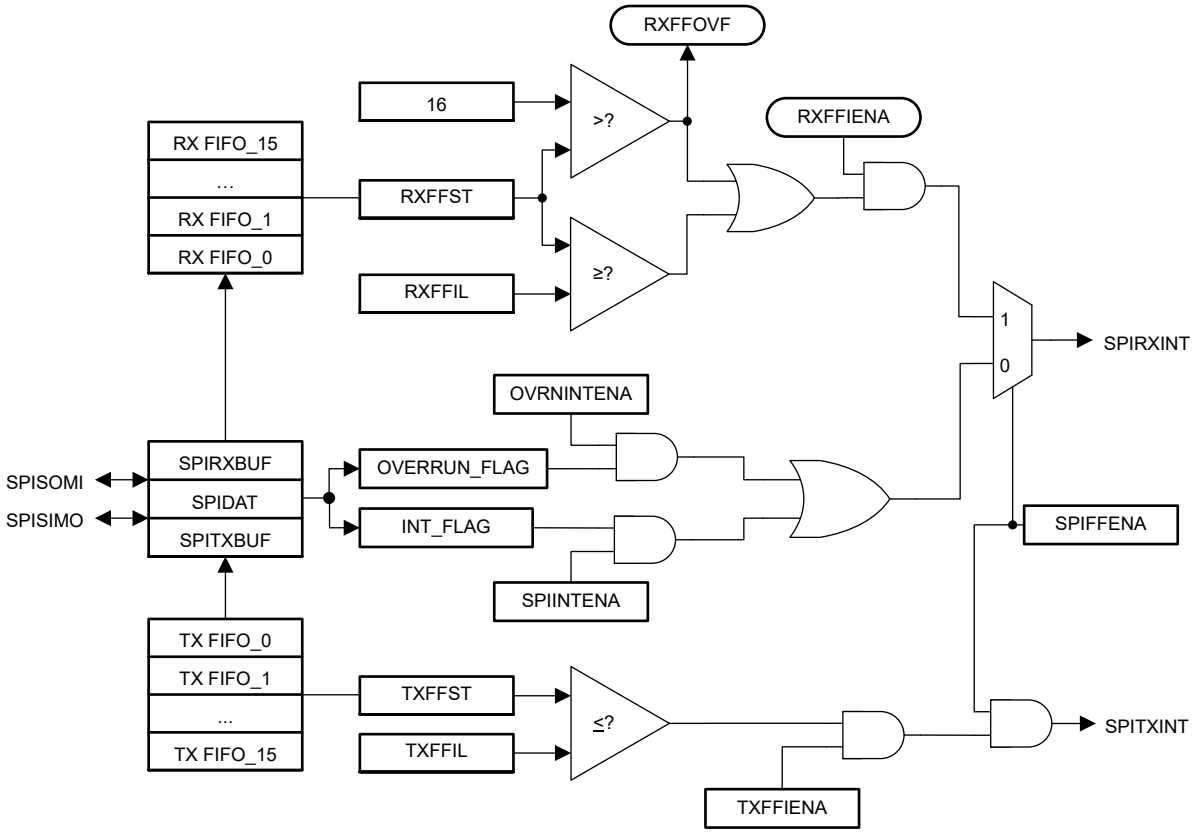


Figure 19-2. SPI Interrupt Flags and Enable Logic Generation

Table 19-2. SPI Interrupt Flag Modes

FIFO Options	SPI Interrupt Source	Interrupt Flags	Interrupt Enables	FIFO Enable (SPIFFENA)	Interrupt Line (1)
SPI without FIFO	Receive overrun	RXOVRN	OVRNINTENA	0	SPIRXINT
	Data receive	SPIINT	SPIINTENA	0	SPIRXINT
	Transmit empty	SPIINT	SPIINTENA	0	SPIRXINT
SPI FIFO mode	FIFO receive	RXFFIL	RXFFIENA	1	SPIRXINT
	Transmit empty	TXFFIL	TXFFIENA	1	SPITXINT

(1) In non-FIFO mode, SPIRXINT is the same name as the SPIINT interrupt in C28x devices.

19.2.4 DMA Support

Both the CPU and DMA have access to the SPI data registers using the internal peripheral bus. This access is limited to 16-bit register reads and writes. Each SPI module can generate two DMA events, SPITXDMA and SPIRXDMA. The DMA events are controlled by configuring the SPIFFTX.TXFFIL and SPIFFRX.RXFFIL appropriately. SPITXDMA activates when TXFFST is less than the interrupt level (TXFFIL). SPIRXDMA activates when RXFFST is greater than or equal to the interrupt level (RXFFIL).

The SPI must have FIFO enhancements enabled for the DMA triggers to be generated.

For more information on configuring the SPI for DMA transfers, refer to Section 19.3.8.

Figure 19-3 is a block diagram showing the DMA trigger generation from the SPI module.

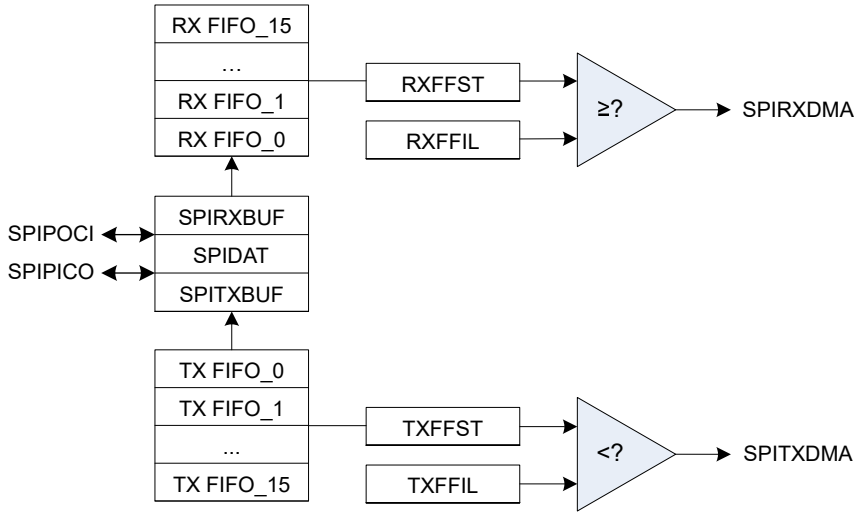


Figure 19-3. SPI DMA Trigger Diagram

19.3 SPI Operation

This section describes the various modes of operation of the SPI. Included are explanations of the operational modes, interrupts, data format, clock sources, and initialization. Typical timing diagrams for data transfers are given.

19.3.1 Introduction to Operation

Figure 19-4 shows typical connections of the SPI for communications between two controllers: a controller and a peripheral.

The controller transfers data by sending the SPICLK signal. For both the peripheral and the controller, data is shifted out of the shift registers on one edge of the SPICLK and latched into the shift register on the opposite SPICLK clock edge. If the CLK_PHASE bit is high, data is transmitted and received a half-cycle before the SPICLK transition. As a result, both controllers send and receive data simultaneously. The application software determines whether the data is meaningful or dummy data. There are three possible methods for data transmission:

- Controller sends data; peripheral sends dummy data.
- Controller sends data; peripheral sends data.
- Controller sends dummy data; peripheral sends data.

The controller can initiate data transfer at any time because the controller controls the SPICLK signal. The software, however, determines how the controller detects when the peripheral is ready to broadcast data.

The SPI operates in controller or peripheral mode. The CONTROLLER_PERIPHERAL bit selects the operating mode and the source of the SPICLK signal.

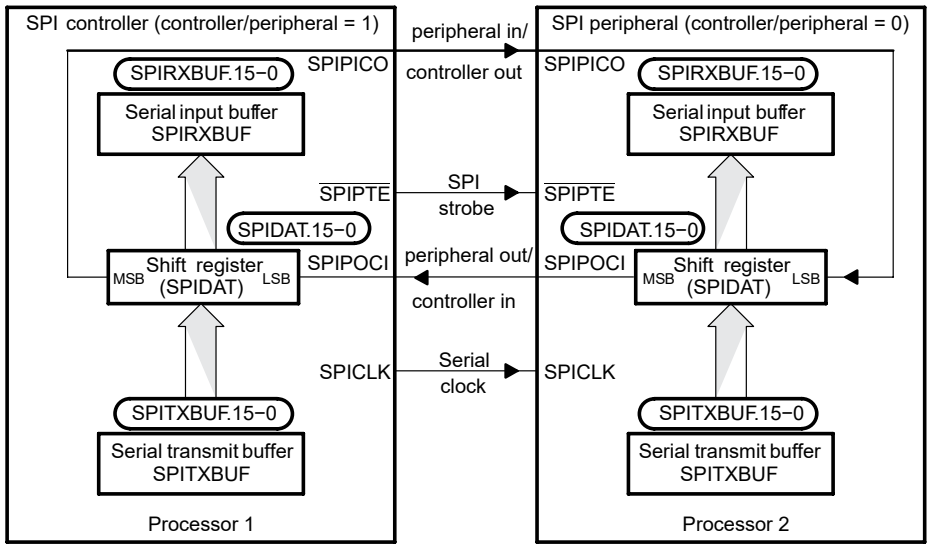


Figure 19-4. SPI Controller/Peripheral Connection

19.3.2 Controller Mode

In controller mode (CONTROLLER_PERIPHERAL = 1), the SPI provides the serial clock on the SPICLK pin for the entire serial communications network. Data is output on the SPIPICO pin and latched from the SPIPOCI pin.

The SPIBRR register determines both the transmit and receive bit transfer rate for the network. SPIBRR can select 125 different data transfer rates.

Data written to SPIDAT or SPITXBUF initiates data transmission on the SPIPICO pin, MSB (most-significant bit) first. Simultaneously, received data is shifted through the SPIPOCI pin into the LSB (least- significant bit) of SPIDAT. When the selected number of bits has been transmitted, the received data is transferred to the SPIRXBUF (buffered receiver) for the CPU to read. Data is stored right-justified in SPIRXBUF.

When the specified number of data bits has been shifted through SPIDAT, the following events occur:

- SPIDAT contents are transferred to SPIRXBUF.
- INT_FLAG bit is set to 1.
- If there is valid data in the transmit buffer SPITXBUF, as indicated by the transmit buffer full flag (BUFFULL_FLAG), this data is transferred to SPIDAT and is transmitted; otherwise, SPICLK stops after all bits have been shifted out of SPIDAT.
- If the SPIINTENA bit is set to 1, an interrupt is asserted.

In a typical application, the $\overline{\text{SPIPTE}}$ pin serves as a chip-enable pin for a SPI peripheral device. This pin is driven low by the controller before transmitting data to the peripheral and is taken high after the transmission is complete.

Figure 19-5 is a block diagram of the SPI in controller mode. The block diagram shows the basic control blocks available in SPI controller mode.

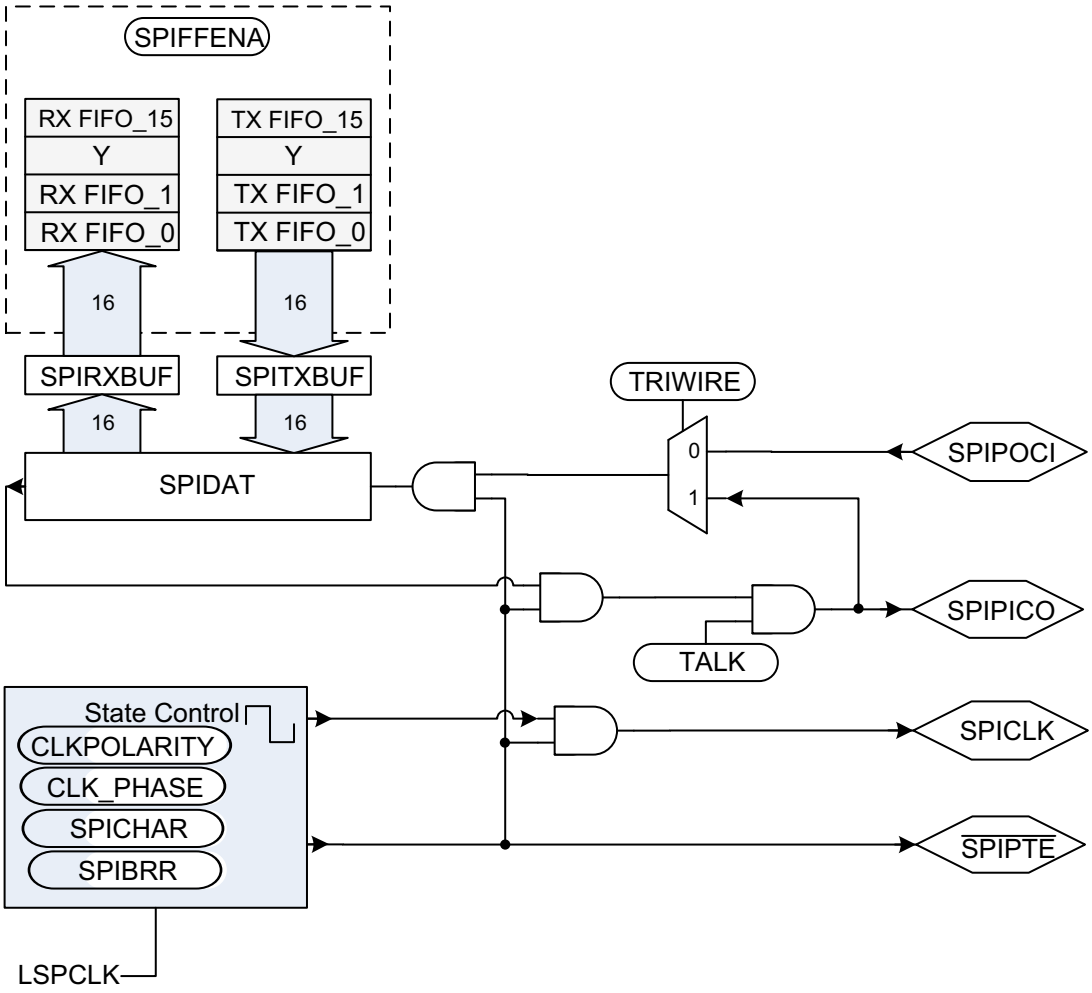


Figure 19-5. SPI Module Controller Configuration

19.3.3 Peripheral Mode

In peripheral mode (CONTROLLER_PERIPHERAL = 0), data shifts out on the SPIPOCI pin and in on the SPIPICO pin. The SPICLK pin is used as the input for the serial shift clock, which is supplied from the external network controller. The transfer rate is defined by this clock. The SPICLK input frequency can be no greater than the LSPCLK frequency divided by 4.

Data written to SPIDAT or SPITXBUF is transmitted to the network when appropriate edges of the SPICLK signal are received from the network controller. A character written to the SPITXBUF register is copied to the SPIDAT register when all bits of the current character in SPIDAT have been shifted out. If no character was previously copied to SPIDAT, then any character written to SPITXBUF is immediately copied to SPIDAT. If a character was previously copied to SPIDAT, any data written to SPITXBUF is not copied to SPIDAT until the current character in SPIDAT has been shifted out. To receive data, the SPI waits for the network controller to send the SPICLK signal and then shifts the data on the SPIPICO pin into SPIDAT. If data is to be transmitted by the peripheral simultaneously, and SPIDAT has not been previously loaded, the character must be written to SPITXBUF before the beginning of the SPICLK signal.

When the TALK bit is cleared, data transmission is disabled, and the output line (SPIPOCI) is put into the high-impedance state. If this occurs while a transmission is active, the current character is completely transmitted even though SPIPOCI is forced into the high-impedance state. This makes sure that the SPI is still able to

receive incoming data correctly. This TALK bit allows many peripheral devices to be tied together on the network, but only one peripheral at a time is allowed to drive the SPIPOCI line.

The $\overline{\text{SPIPTE}}$ pin operates as the peripheral-select pin. An active-low signal on the $\overline{\text{SPIPTE}}$ pin allows the peripheral SPI to transfer data to the serial data line; an inactive-high signal causes the peripheral SPI serial shift register to stop and the serial output pin to be put into the high-impedance state. This allows many peripheral devices to be tied together on the network, although only one peripheral device is selected at a time.

Figure 19-6 is a block diagram of the SPI in peripheral mode. The block diagram shows the basic control blocks available in SPI peripheral mode.

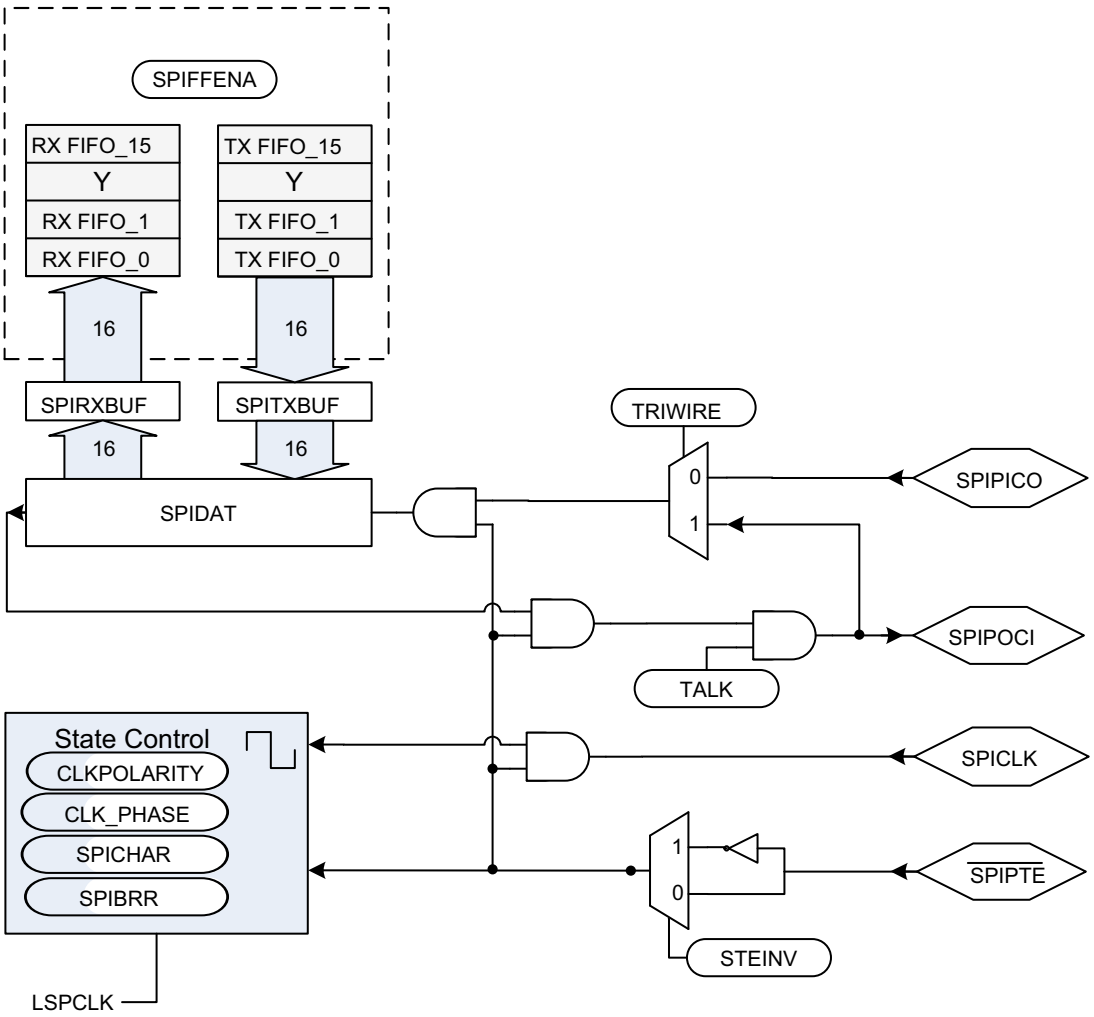


Figure 19-6. SPI Module Peripheral Configuration

19.3.4 Data Format

The 4-bit SPICHAR register field specifies the number of bits in the data character (1 to 16). This information directs the state control logic to count the number of bits received or transmitted to determine when a complete character has been processed.

The following statements apply to characters with fewer than 16 bits:

- Data must be left-justified when written to SPIDAT and SPITXBUF.
- Data read back from SPIRXBUF is right-justified.
- SPIRXBUF contains the most recently received character, right-justified, plus any bits that remain from previous transmissions that have been shifted to the left (shown in Example 19-1).

Example 19-1. Transmission of Bit from SPIRXBUF

Conditions:

1. Transmission character length = 1 bit (specified in SPICHAR bits)
2. The current value of SPIDAT = 737Bh

SPIDAT (before transmission)																
	0	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1
SPIDAT (after transmission)																
(TXed) 0 ←	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	x[1] ← (RXed)
SPIRXBUF (after transmission)																
	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	x[1]

(1) x = 1, if SPIPOCI data is high; x = 0, if SPIPOCI data is low; controller mode is assumed.

19.3.5 Baud Rate Selection

The SPI module supports 125 different baud rates and four different clock schemes. Depending on whether the SPI clock is in peripheral or controller mode, the SPICLK pin can receive an external SPI clock signal or provide the SPI clock signal, respectively.

- In the peripheral mode, the SPI clock is received on the SPICLK pin from the external source and can be no greater than the LSPCLK frequency divided by 4.
- In the controller mode, the SPI clock is generated by the SPI and is output on the SPICLK pin and can be no greater than the LSPCLK frequency divided by 4.

Note

The baud rate must be configured to not exceed the maximum rated GPIO toggle frequency. Refer to the device data sheet for the maximum GPIO toggle frequency.

Example 19-2 shows how to determine the SPI baud rates.

Example 19-3 shows how to calculate the baud rate of the SPI module in standard SPI mode (HS_MODE = 0).

Example 19-2. Baud Rate Determination

For SPIBRR = 3 to 127:

$$\text{SPI Baud Rate} = \frac{\text{LSPCLK}}{(\text{SPIBRR} + 1)}$$

For SPIBRR = 0, 1, or 2:

$$\text{SPI Baud Rate} = \frac{\text{LSPCLK}}{4}$$

where:

LSPCLK = Low-speed peripheral clock frequency of the device

SPIBRR = Contents of the SPIBRR in the controller SPI device

To determine what value to load into SPIBRR, you must know the device system clock (LSPCLK) frequency (that is device-specific) and the baud rate at which you are operating.

Example 19-3. Baud Rate Calculation in Non-High Speed Mode (HS_MODE = 0)

$$SPI\ Baud\ Rate = \frac{LSPCLK}{SPIBRR + 1'} \quad LSPCLK = 50\ MHz$$

$$= \frac{50 \times 10^6}{3 + 1}$$

$$= 12.5\ Mbps$$

19.3.6 SPI Clocking Schemes

The clock polarity select bit (CLKPOLARITY) and the clock phase select bit (CLK_PHASE) control four different clocking schemes on the SPICLK pin. CLKPOLARITY selects the active edge, either rising or falling, of the clock. CLK_PHASE selects a half-cycle delay of the clock. The four different clocking schemes are:

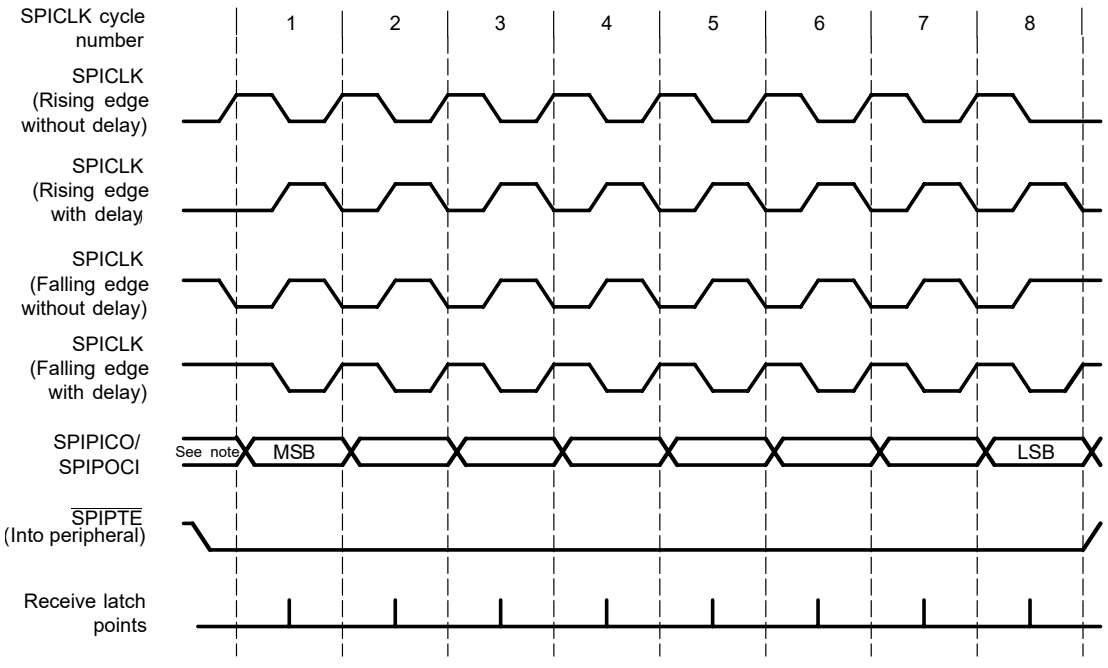
- Falling Edge Without Delay. The SPI transmits data on the falling edge of the SPICLK and receives data on the rising edge of the SPICLK.
- Falling Edge With Delay. The SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
- Rising Edge Without Delay. The SPI transmits data on the rising edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
- Rising Edge With Delay. The SPI transmits data one half-cycle ahead of the rising edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.

The selection procedure for the SPI clocking scheme is shown in Table 19-3. Examples of these four clocking schemes relative to transmitted and received data are shown in Figure 19-7.

Table 19-3. SPI Clocking Scheme Selection Guide

SPICLK Scheme	CLKPOLARITY(1)	CLK_PHASE(1)
Rising edge without delay	0	0
Rising edge with delay	0	1
Falling edge without delay	1	0
Falling edge with delay	1	1

(1) The description of CLK_PHASE and CLKPOLARITY differs between manufacturers. For proper operation, select the desired waveform to determine the clock phase and clock polarity settings.



Note: Previous data bit

Figure 19-7. SPICLK Signal Options

The SPICLK symmetry is retained only when the result of (SPIBRR + 1) is an even value. When (SPIBRR + 1) is an odd value and SPIBRR is greater than 3, SPICLK becomes asymmetrical. The low pulse of SPICLK is one LSPCLK cycle longer than the high pulse when CLKPOLARITY bit is clear (0). When CLKPOLARITY bit is set to 1, the high pulse of the SPICLK is one LSPCLK cycle longer than the low pulse, as shown in Figure 19-8.

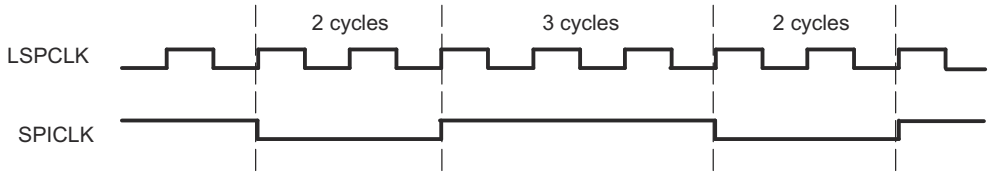


Figure 19-8. SPI: SPICLK-LSPCLK Characteristic when (BRR + 1) is Odd, BRR > 3, and CLKPOLARITY = 1

19.3.7 SPI FIFO Description

The following steps explain the FIFO features and help with programming the SPI FIFOs:

1. **Reset.** At reset the SPI powers up in standard SPI mode and the FIFO function is disabled. The FIFO registers SPIFFTX, SPIFFRX and SPIFFCT remain inactive.
2. **Standard SPI.** The standard 28x SPI mode works with SPIINT/SPIRXINT as the interrupt source.
3. **Mode change.** FIFO mode is enabled by setting the SPIFFENA bit to 1 in the SPIFFTX register. SPIRST can reset the FIFO mode at any stage of the operation.
4. **Active registers.** All the SPI registers and SPI FIFO registers SPIFFTX, SPIFFRX, and SPIFFCT are active.
5. **Interrupts.** FIFO mode has two interrupts: one for the transmit FIFO, SPITXINT; one for the receive FIFO, SPIRXINT. SPIRXINT is the common interrupt for SPI FIFO receive, receive error and receive FIFO overflow conditions. The single SPIINT for both transmit and receive sections of the standard SPI are disabled and this interrupt is serviced as SPI receive FIFO interrupt. For more information, refer to Section 19.2.3.
6. **Buffers.** Transmit and receive buffers are each supplemented with a 16-word FIFO. The one-word transmit buffer (SPITXBUF) of the standard SPI functions as a transition buffer between the transmit FIFO and shift register. The one-word transmit buffer is loaded from transmit FIFO only after the last bit of the shift register is shifted out.
7. **Delayed transfer.** The rate at which transmit words in the FIFO are transferred to transmit shift register is programmable. The SPIFFCT register bits (7−0) FFTXDLY7−FFTXDLY0 define the delay between the word transfer. The delay is defined in number SPI serial clock cycles. The 8-bit register can define a minimum delay of 0 SPICLK cycles and a maximum of 255 SPICLK cycles. With zero delay, the SPI module can transmit data in continuous mode with the FIFO words shifting out back to back. With the 255-clock delay, the SPI module can transmit data in a maximum delayed mode with the FIFO words shifting out with a delay of 255 SPICLK cycles between each words. The programmable delay facilitates glueless interface to various slow SPI peripherals, such as EEPROMs, ADC, DAC, and so on.
8. **FIFO status bits.** Both transmit and receive FIFOs have status bits TXFFST or RXFFST that define the number of words available in the FIFOs at any time. The transmit FIFO reset bit (TXFIFO) and receive reset bit (RXFIFO) reset the FIFO pointers to zero when these bits are set to 1. The FIFOs resume operation from start once these bits are cleared to zero.
9. **Programmable interrupt levels.** Both transmit and receive FIFOs can generate CPU interrupts and DMA triggers. The transmit interrupt (SPITXINT) is generated whenever the transmit FIFO status bits (TXFFST) match (less than or equal to) the interrupt trigger level bits (TXFFIL). The receive interrupt (SPIRXINT) is generated whenever the receive FIFO status bits (RXFFST) match (greater than or equal to) the interrupt trigger level RXFFIL. This provides a programmable interrupt trigger for transmit and receive sections of the SPI. The default value for these trigger level bits is 0x11111 for receive FIFO and 0x00000 for transmit FIFO, respectively.

19.3.8 SPI DMA Transfers

19.3.8.1 Transmitting Data Using SPI with DMA

When using the DMA with the TX FIFO, the DMA Burst Size (DMA_BURST_SIZE) must be no greater than 16 - TXFFIL, to prevent the DMA from writing to an already full FIFO. This leads to data loss and is not recommended.

For complete data transmission, follow these steps:

1. Calculate the total number of words to be transmitted. [NUM_WORDS]
2. Decide the transmit FIFO level. [TXFFIL]
3. Calculate the total number of DMA transfers. [DMA_TRANSFER_SIZE]
4. Calculate the size of the DMA Burst. [DMA_BURST_SIZE]
5. Configure DMA using calculated values.
6. Configure SPI with FIFO using the calculated values.

To transfer 128 words to SPI using the DMA:

NUM_WORDS: 128

TXFFIL: 8

DMA_TRANSFER_SIZE: (NUM_WORDS /(16 - TXFFIL)) - 1 = (128/8) - 1 = 15 (16 transfers)

DMA_BURST_SIZE: (16 - TXFFIL) - 1 = (16 - 8) - 1 = 7 (8 words per burst)

Note

Avoid setting TXFFIL to 0h or 10h to make sure of proper DMA configuration.

19.3.8.2 Receiving Data Using SPI with DMA

When using the DMA with the RX FIFO, the DMA Burst Size (BURST_SIZE) must be no greater than RXFFIL to prevent the DMA from reading from an empty FIFO. To make sure that the DMA correctly receives all data from the RX FIFO, the DMA Burst Size can equal RXFFIL and also be an integer divisor of the total number of SPI transmissions.

For complete data reception, follow these steps:

1. Calculate the total number of words to be received. [NUM_WORDS]
2. Calculate the necessary FIFO level [RXFFIL]
3. Calculate the total number of DMA transfers. [DMA_TRANSFER_SIZE]
4. Calculate the size of the DMA Burst. [DMA_BURST_SIZE]
5. Configure DMA using the calculated values.
6. Configure SPI with FIFO using the calculated values.

To receive 200 words from SPI using the DMA:

NUM_WORDS = 200

RXFFIL: 4

DMA_TRANSFER_SIZE: (NUM_WORDS /RXFFIL) - 1 = (200/4) - 1 = 49 (50 transfers)

DMA_BURST_SIZE = RXFFIL - 1 = 3 (4 words per burst)

Note

Avoid setting RXFFIL to 0h to make sure proper DMA configuration.

19.3.9 SPI High-Speed Mode

The SPI module is capable of reaching full-duplex communication speeds up to LSPCLK/4 (where LSPCLK equals SYSCLK). For the maximum rated speed, refer to the device data sheet.

To achieve the maximum full-duplex speeds, the following restrictions are placed on the design:

- Single controller to single peripheral configuration is supported.
- Loading on the pins must not exceed the value stated in the device data sheet.

When configuring the GPIOs to support high-speed mode, refer to Section 19.2.2.1 for more information.

19.3.10 SPI 3-Wire Mode Description

SPI 3-wire mode allows for SPI communication over three pins instead of the normal four pins.

In controller mode, if the TRIWIRE bit is set, enabling 3-wire SPI mode, SPIPICOx becomes the bi-directional SPICOCIx (SPI controller out, controller in) pin, and SPIPOCIx is no longer used by the SPI. In peripheral mode, if the TRIWIRE bit is set, SPIPOCIx becomes the bi-directional SPIPIPOx (SPI peripheral in, peripheral out) pin, and SPIPICOx is no longer used by the SPI.

Table 19-4 indicates the pin function differences between 3-wire and 4-wire SPI mode for a controller and peripheral SPI.

Table 19-4. 4-wire versus 3-wire SPI Pin Functions

4-wire SPI	3-wire SPI (Controller)	3-wire SPI (Peripheral)
SPICLKx	SPICLKx	SPICLKx
SPISTEx	SPISTEx	SPISTEx
SPIPICOx	SPICOCIx	Free
SPIPOCIx	Free	SPIPIPOx

Because in 3-wire mode, the receive and transmit paths within the SPI are connected, any data transmitted by the SPI module is also received. The application software must take care to perform a dummy read to clear the SPI data register of the additional received data.

The TALK bit plays an important role in 3-wire SPI mode. The bit must be set to transmit data and cleared prior to reading data. In controller mode, to initiate a read, the application software must write dummy data to the SPI data register (SPIDAT or SPIRXBUF) while the TALK bit is cleared (no data is transmitted out the SPICOCI pin) before reading from the data register.

Figure 19-9 and Figure 19-10 illustrate 3-wire controller and peripheral mode.

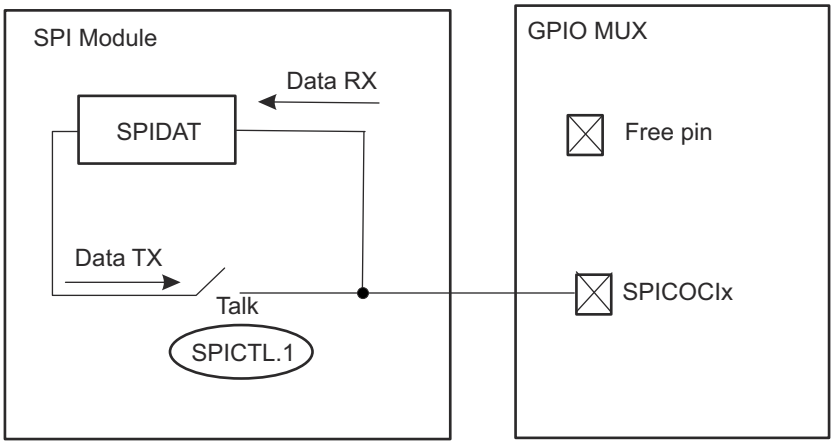


Figure 19-9. SPI 3-wire Controller Mode

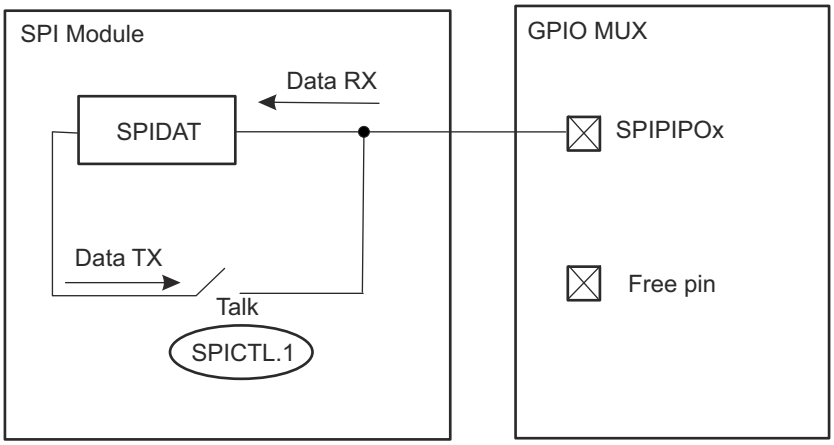


Figure 19-10. SPI 3-wire Peripheral Mode

Table 19-5 indicates how data is received or transmitted in the various SPI modes while the TALK bit is set or cleared.

Table 19-5. 3-Wire SPI Pin Configuration

Pin Mode	SPIPRI[TRIWIRE]	SPICTL[TALK]	SPIPICO	SPIPOCI
		Controller Mode		
4-wire	0	X	TX	RX
3-pin mode	1	0	RX	Disconnect from SPI
		1	TX/RX	
		Peripheral Mode		
4-wire	0	X	RX	TX
3-pin mode	1	0	Disconnect from SPI	RX
		1		TX/RX

19.4 Programming Procedure

This section describes the procedure for configuring the SPI for the various modes of operation.

19.4.1 Initialization Upon Reset

A system reset forces the SPI peripheral into the following default configuration:

- Unit is configured as a peripheral module (CONTROLLER_PERIPHERAL = 0)
- Transmit capability is disabled (TALK = 0)
- Data is latched at the input on the falling edge of the SPICLK signal
- Character length is assumed to be one bit
- SPI interrupts are disabled
- Data in SPIDAT is reset to 0000h

19.4.2 Configuring the SPI

This section describes the procedure in which to configure the SPI module for operation. To prevent unwanted and unforeseen events from occurring during or as a result of initialization changes, clear the SPISWRESET bit before making initialization changes, and then set this bit after initialization is complete. While the SPI is held in reset (SPISWRESET = 0), configuration can be changed in any order. The following list shows the SPI configuration procedure in a logical order. However, the SPI registers can be written with single 16-bit writes, so the order is not required with the exception of SPISWRESET.

Note

Do not change the SPI configuration when communication is in progress.

To change the SPI configuration:

1. Clear the SPI Software Reset bit (SPISWRESET) to 0 to force the SPI to the reset state.
2. Configure the SPI as desired:
   - Select either controller or peripheral mode (CONTROLLER_PERIPHERAL).
   - Choose SPICLK polarity and phase (CLKPOLARITY and CLK_PHASE).
   - Set the desired baud rate (SPIBRR).
   - Enable high-speed mode, if desired (HS_MODE).
   - Set the SPI character length (SPICHAR).
   - Clear the SPI Flags (OVERRUN_FLAG, INT_FLAG).
   - Enable $\overline{\text{SPIPTE}}$ inversion (STEINV), if needed.
   - Enable 3-wire mode (TRIWIRE), if needed.
   - If using FIFO enhancements:
     - Enable the FIFO enhancements (SPIFFENA).
     - Clear the FIFO Flags (TXFFINTCLR, RXFFOVFCLR, and RXFFINTCLR).
     - Release transmit and receive FIFO resets (TXFIFO and RXFIFORESET).
     - Release SPI FIFO channels from reset (SPIRST).
3. If interrupts are used:
   - In non-FIFO mode, enable the receiver overrun and/or SPI interrupts (OVERRUNINTENA and SPIINTENA).
   - In FIFO mode, set the transmit and receive interrupt levels (TXFFIL and RXFFIL) then enable the interrupts (TXFFIENA and RXFFIENA).
4. Set SPISWRESET to 1 to release the SPI from the reset state.

19.4.3 Configuring the SPI for High-Speed Mode

To achieve the maximum rated speeds, the following settings must be made. This example assumes that the device is operating at 100MHz.

Set LSPCLK equal to SYSCLK:

```
ClkCfgRegs.LOSPCP.bit.LSPCLKDIV = 0;
```

Select the appropriate Pin Mux options in GPIO_CTRL_REGS.

During the SPI configuration procedure:

Set HS_MODE to 1.

```
SpiaRegs.SPICCR.bit.HS_MODE = 0x1;
```

Set SPIBRR to 3. SPICLK = LSPCLK/(SPIBRR+1) = 25MHz

```
SpiaRegs.SPIBRR = 0x3;
```

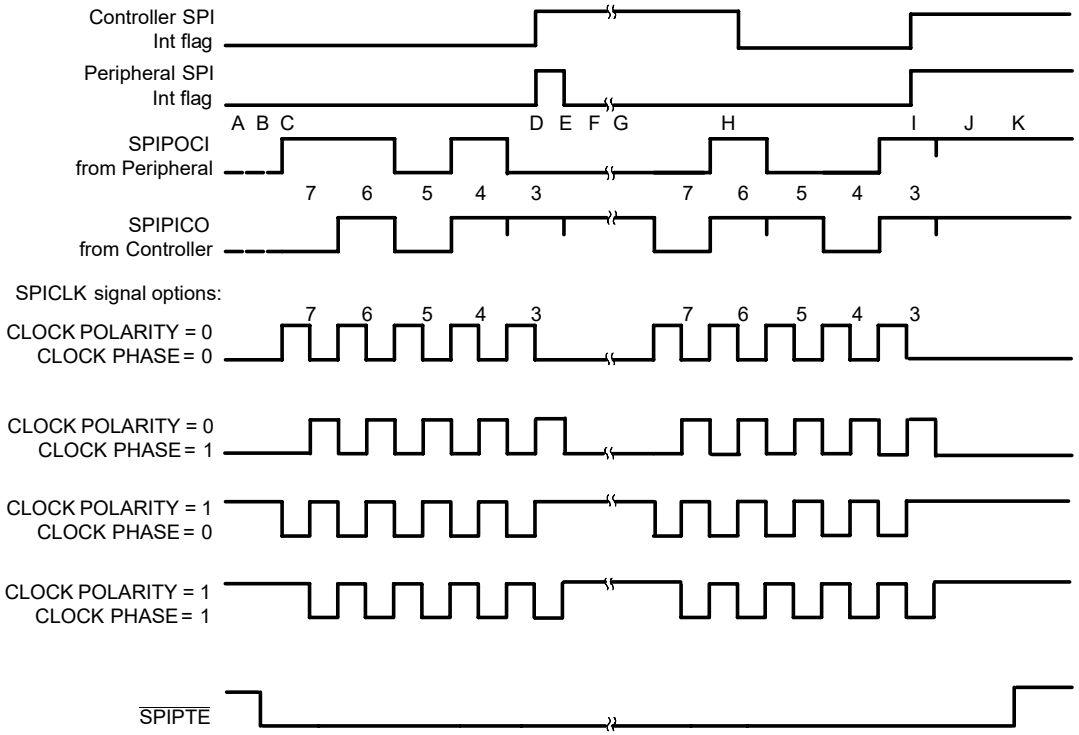
There are no other differences in the configuration from normal SPI operation. Sending and receiving data, DMA operation, and interrupts operates without change.

19.4.4 Data Transfer Example

The timing diagram shown in Figure 19-11 shows an SPI data transfer between two devices using a character length of five bits with the SPICLK being symmetrical.

The timing diagram with SPICLK asymmetrical (Figure 19-8) shares similar characterizations with Figure 19-11 except that the data transfer is one LSPCLK cycle longer per bit during the low pulse (CLKPOLARITY = 0) or during the high pulse (CLKPOLARITY = 1) of the SPICLK.

Figure 19-11 is applicable for 8-bit SPI only and is not for C28x devices that are capable of working with 16-bit data. The figure is shown for illustrative purposes only.



A. Peripheral writes 0D0h to SPIDAT and waits for the controller to shift out the data.
B. Controller sets the peripheral $\overline{\text{SPIPTE}}$ signal low (active).
C. Controller writes 058h to SPIDAT, which starts the transmission procedure.
D. First byte is finished and sets the interrupt flags.
E. Peripheral reads 0Bh from the SPIRXBUF (right-justified).
F. Peripheral writes 04Ch to SPIDAT and waits for the controller to shift out the data.
G. Controller writes 06Ch to SPIDAT, which starts the transmission procedure.
H. Controller reads 01Ah from the SPIRXBUF (right-justified).
I. Second byte is finished and sets the interrupt flags.
J. Controller reads 89h and the peripheral reads 8Dh from the respective SPIRXBUF. After the user software masks off the unused bits, the controller receives 09h and the peripheral receives 0Dh.
K. Controller clears the peripheral $\overline{\text{SPIPTE}}$ signal high (inactive).

Figure 19-11. Five Bits per Character

19.4.5 SPI 3-Wire Mode Code Examples

In addition to the normal SPI initialization, to configure the SPI module for 3-wire mode, the TRIWIRE bit (SPIPRI.0) must be set to 1. After initialization, there are several considerations to take into account when transmitting and receiving data in 3-wire controller and peripheral mode. The following examples demonstrate these considerations.

In 3-wire controller mode, SPICLKx, $\overline{\text{SPIPTE}}$x, and SPIPICOx pins must be configured as SPI pins (SPIPOCIx pin can be configured as non-SPI pin). When the controller transmits, the controller receives the data the controller transmits (because SPIPICOx and SPIPOCIx are connected internally in 3-wire mode). Therefore, the junk data received must be cleared from the receive buffer every time data is transmitted.

Example 19-4. 3-Wire Controller Mode Transmit

```
Uint16 data;
Uint16 dummy;
     SpiaRegs.SPICTL.bit.TALK = 1;              // Enable Transmit path
     SpiaRegs.SPITXBUF = data; // Controller transmits data
     while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {} // Waits until data rx'd
     dummy = SpiaRegs.SPIRXBUF;                 // Clears junk data because
                                                // rx'd same data as tx'd
```

To receive data in 3-wire controller mode, the controller must clear the TALK (SPICTL.1) bit to 0 to close the transmit path and then transmit dummy data to initiate the transfer from the peripheral. Because the TALK bit is 0, unlike in transmit mode, the controller dummy data does not appear on the SPIPICOx pin, and the controller does not receive the dummy data. Instead, the data from the peripheral is received by the controller.

Example 19-5. 3-Wire Controller Mode Receive

```
Uint16 rdata;
Uint16 dummy;
     SpiaRegs.SPICTL.bit.TALK = 0;              // Disable Transmit path
     SpiaRegs.SPITXBUF = dummy;                 // Send dummy to start tx
     // NOTE: because TALK = 0, data does not tx onto SPIPICOA pin
     while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {} // Wait until data received
     rdata = SpiaRegs.SPIRXBUF;                 // Controller reads data
```

In 3-wire peripheral mode, SPICLKx, SPISTEx, and SPIPOCIx pins must be configured as SPI pins (SPIPICOx pin can be configured as non-SPI pin). Like in controller mode, when transmitting, the peripheral receives the data transmitted and must clear this junk data from the receive buffer.

Example 19-6. 3-Wire Peripheral Mode Transmit

```
Uint16 data;
Uint16 dummy;
     SpiaRegs.SPICTL.bit.TALK = 1;              // Enable Transmit path
     SpiaRegs.SPITXBUF = data;                  // Peripheral transmits data
     while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {} // Wait until data rx'd
     dummy = SpiaRegs.SPIRXBUF;                 // Clears junk data
```

As in 3-wire controller mode, the TALK bit must be cleared to 0. Otherwise, the peripheral receives data normally.

Example 19-7. 3-Wire Peripheral Mode Receive

```
Uint16 rdata;
    SpiaRegs.SPICTL.bit.TALK = 0;              // Disable Transmit path
    while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {} // Waits until data rx'd
    rdata = SpiaRegs.SPIRXBUF;                 // Peripheral reads data
```

19.4.6 SPI STEINV Bit in Digital Audio Transfers

On those devices with two SPI modules, enabling the STEINV bit on one of the SPI modules allows the pair of SPIs to receive both left and right-channel digital audio data in peripheral mode. The SPI module that receives a normal active-low $\overline{\text{SPIPTE}}$ signal stores right-channel data, and the SPI module that receives an inverted active-high $\overline{\text{SPIPTE}}$ signal stores left-channel data from the controller. To receive digital audio data from a digital audio interface receiver, the SPI modules can be connected as shown in Figure 19-12.

Note

This configuration is only applicable to peripheral mode (CONTROLLER_PERIPHERAL = 0). When the SPI is configured as controller (CONTROLLER_PERIPHERAL = 1), the STEINV bit has no effect on the $\overline{\text{SPIPTE}}$ pin.

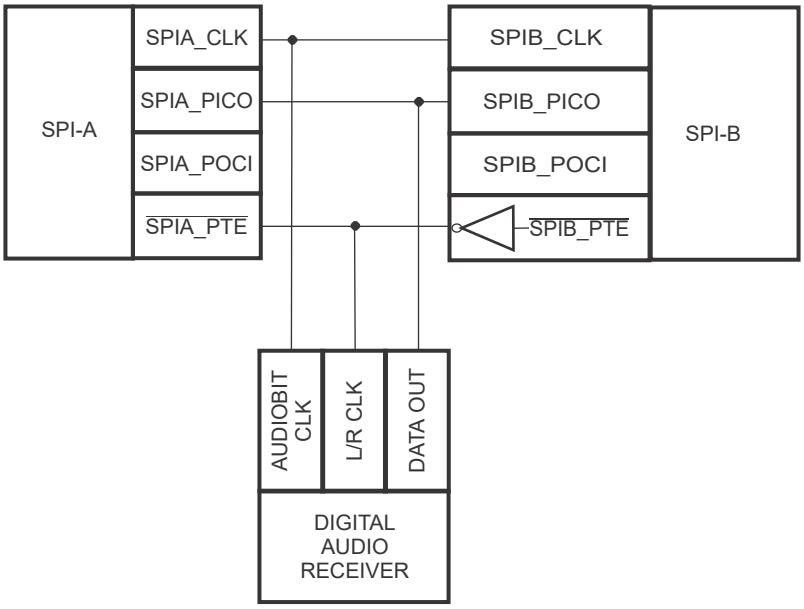


Figure 19-12. SPI Digital Audio Receiver Configuration Using Two SPIs

Standard C28x SPI timing requirements limit the number of digital audio interface formats supported using the 2-SPI configuration with the STEINV bit. See the device data sheet electrical specifications for SPI timing requirements. With the SPI clock phase configured such that the CLKPOLARITY bit is 0 and the CLK_PHASE bit is 1 (data latched on rising edge of clock), standard right-justified digital audio interface data format is supported as shown in Figure 19-13.

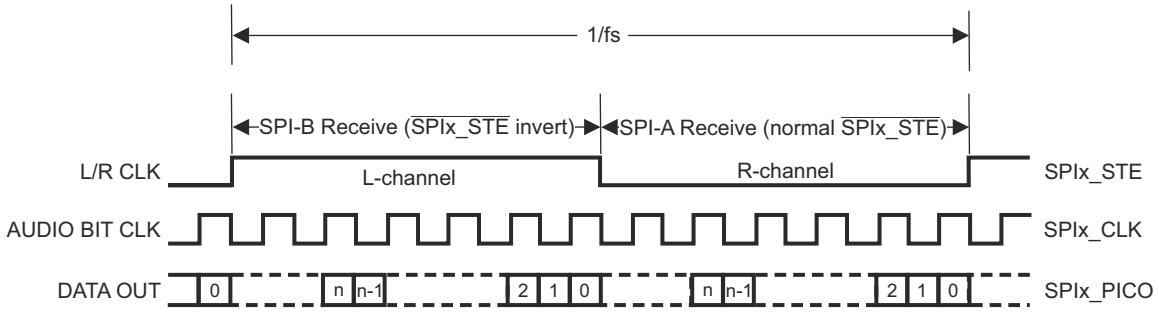


Figure 19-13. Standard Right-Justified Digital Audio Data Format

19.5 Software

19.5.1 SPI Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/spi

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

19.5.1.1 SPI Digital Loopback

FILE: spi_ex1_loopback.c

This program uses the internal loopback test mode of the SPI module. This is a very basic loopback that does not use the FIFOs or interrupts. A stream of data is sent and then compared to the received stream. The pinmux and SPI modules are configure through the sysconfig file.

The sent data looks like this:
0000 0001 0002 0003 0004 0005 0006 0007 .... FFFE FFFF 0000

This pattern is repeated forever.

This example project has support for migration across our C2000 device families. If you are wanting to build this project from launchpad or controlCARD, please specify in the .syscfg file the board you're using. At any time you can select another device to migrate this example. *External Connections*

- None

Watch Variables

- *sData* - Data to send
- *rData* - Received data

19.5.1.2 SPI Digital Loopback with FIFO Interrupts

FILE: spi_ex2_loopback_fifo_interrupts.c

This program uses the internal loopback test mode of the SPI module. Both the SPI FIFOs and their interrupts are used.

A stream of data is sent and then compared to the received stream. The sent data looks like this:
0000 0001
0001 0002
0002 0003
....
FFFE FFFF
FFFF 0000
etc..
This pattern is repeated forever.

This example project has support for migration across our C2000 device families. If you are wanting to build this project from launchpad or controlCARD, please specify in the .syscfg file the board you're using. At any time you can select another device to migrate this example. *External Connections*

- None

Watch Variables

- *sData* - Data to send
- *rData* - Received data
- *rDataPoint* - Used to keep track of the last position in the receive stream for error checking

19.5.1.3 SPI Digital Loopback with DMA

FILE: spi_ex5_loopback_dma.c

This program uses the internal loopback test mode of the SPI module. Both DMA interrupts and the SPI FIFOs are used. When the SPI transmit FIFO has enough space (as indicated by its FIFO level interrupt signal), the DMA will transfer data from global variable sData into the FIFO. This will be transmitted to the receive FIFO via the internal loopback.

When enough data has been placed in the receive FIFO (as indicated by its FIFO level interrupt signal), the DMA will transfer the data from the FIFO into global variable rData.

When all data has been placed into rData, a check of the validity of the data will be performed in one of the DMA channels' ISRs.

This example project has support for migration across our C2000 device families. If you are wanting to build this project from launchpad or controlCARD, please specify in the .syscfg file the board you're using. At any time you can select another device to migrate this example. *External Connections*

- None

Watch Variables

- *sData* - Data to send
- *rData* - Received data

19.5.1.4 SPI EEPROM

FILE: spi_ex6_eeprom.c

This program will write 8 bytes to EEPROM and read them back. The device communicates with the EEPROM via SPI and specific opcodes. This example is written to work with the SPI Serial EEPROM AT25128/256. Note: SPI character length is configured to 8 bits in SysConfig, and not changed throughout the execution of the program. Runtime updation of character length when CS pin is not controlled by the SPI module can lead to unpredictable behavior.

External Connections

- Connect external SPI EEPROM
- Connect GPIO2 (PICO) to external EEPROM SI pin
- Connect GPIO5 (POCI) to external EEPROM SO pin
- Connect GPIO3 (CLK) to external EEPROM SCK pin
- Connect GPIO12 (CS) to external EEPROM CS pin
- Connect the external EEPROM VCC and GND pins

Watch Variables

- writeBuffer - Data that is written to external EEPROM
- readBuffer - Data that is read back from EEPROM
- error - Error count

19.5.1.5 SPI DMA EEPROM

FILE: spi_ex7_eeprom_dma.c

This program will write 8 bytes to EEPROM and read them back. The device communicates with the EEPROM via SPI using DMA and specific opcodes. This example is written to work with the SPI Serial EEPROM AT25128/256. Note: SPI character length is configured to 8 bits in SysConfig, and not changed throughout the execution of the program. Runtime updation of character length when CS pin is not controlled by the SPI module can lead to unpredictable behaviour

External Connections

- Connect external SPI EEPROM
- Connect GPIO2 (PICO) to external EEPROM SI pin
- Connect GPIO5 (POCI) to external EEPROM SO pin
- Connect GPIO3 (CLK) to external EEPROM SCK pin
- Connect GPIO12 (CS) to external EEPROM CS pin

- Connect the external EEPROM VCC and GND pins

Watch Variables

- writeBuffer - Data that is written to external EEPROM
- SPI_DMA_Handle.RXdata - Data that is read back from EEPROM when number of received bytes is less than 4
- SPI_DMA_Handle.pSPIRXDMA->pbuffer - Start address of received data from EEPROM
- error - Error count

19.6 SPI Registers

This Section describes the SPI Registers.

19.6.1 SPI Base Address Table

Table 19-6. SPI Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
SpiaRegs	SPI_REGS	SPIA_BASE	0x0000_6100	YES	YES

19.6.2 SPI_REGS Registers

Table 19-7 lists the memory-mapped registers for the SPI_REGS registers. All register offset addresses not listed in Table 19-7 should be considered as reserved locations and the register contents should not be modified.

Table 19-7. SPI_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	SPICCR	SPI Configuration Control Register	
1h	SPICTL	SPI Operation Control Register	
2h	SPISTS	SPI Status Register	
4h	SPIBRR	SPI Baud Rate Register	
6h	SPIRXEMU	SPI Emulation Buffer Register	
7h	SPIRXBUF	SPI Serial Input Buffer Register	
8h	SPITXBUF	SPI Serial Output Buffer Register	
9h	SPIDAT	SPI Serial Data Register	
Ah	SPIFFTX	SPI FIFO Transmit Register	
Bh	SPIFFRX	SPI FIFO Receive Register	
Ch	SPIFFCT	SPI FIFO Control Register	
Fh	SPIPRI	SPI Priority Control Register	

Complex bit access types are encoded to fit into small table cells. Table 19-8 shows the codes that are used for access types in this section.

Table 19-8. SPI_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
RC	R C	Read to Clear
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
Reset or Default Value		
-n		Value after reset or the default value

19.6.2.1 SPICCR Register (Offset = 0h) [Reset = 0000h]

SPICCR is shown in Figure 19-14 and described in Table 19-9.

Return to the Summary Table.

SPICCR controls the setup of the SPI for operation.

Figure 19-14. SPICCR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SPISWRESET	CLKPOLARITY	HS_MODE	SPILBK	SPICHAR			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			

Table 19-9. SPICCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	SPISWRESET	R/W	0h	SPI Software Reset When changing configuration, you should clear this bit before the changes and set this bit before resuming operation. Reset type: SYSRSn 0h (R/W) = Initializes the SPI operating flags to the reset condition. Specifically, the RECEIVER OVERRUN Flag bit (SPISTS.7), the SPI INT FLAG bit (SPISTS.6), and the TXBUF FULL Flag bit (SPISTS.5) are cleared. SPIPTE will become inactive. SPICLK will be immediately driven to 0 regardless of the clock polarity. The SPI configuration remains unchanged. 1h (R/W) = SPI is ready to transmit or receive the next character. When the SPI SW RESET bit is a 0, a character written to the transmitter will not be shifted out when this bit is set. A new character must be written to the serial data register. SPICLK will be returned to its inactive state one SPICLK cycle after this bit is set.
6	CLKPOLARITY	R/W	0h	Shift Clock Polarity This bit controls the polarity of the SPICLK signal. CLOCK POLARITY and POLARITY CLOCK PHASE (SPICTL.3) control four clocking schemes on the SPICLK pin. Reset type: SYSRSn 0h (R/W) = Data is output on rising edge and input on falling edge. When no SPI data is sent, SPICLK is at low level. The data input and output edges depend on the value of the CLOCK PHASE bit (SPICTL.3) as follows: - CLOCK PHASE = 0: Data is output on the rising edge of the SPICLK signal. Input data is latched on the falling edge of the SPICLK signal. - CLOCK PHASE = 1: Data is output one half-cycle before the first rising edge of the SPICLK signal and on subsequent falling edges of the SPICLK signal. Input data is latched on the rising edge of the SPICLK signal. 1h (R/W) = Data is output on falling edge and input on rising edge. When no SPI data is sent, SPICLK is at high level. The data input and output edges depend on the value of the CLOCK PHASE bit (SPICTL.3) as follows: - CLOCK PHASE = 0: Data is output on the falling edge of the SPICLK signal. Input data is latched on the rising edge of the SPICLK signal. - CLOCK PHASE = 1: Data is output one half-cycle before the first falling edge of the SPICLK signal and on subsequent rising edges of the SPICLK signal. Input data is latched on the falling edge of the SPICLK signal.

Table 19-9. SPICCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	HS_MODE	R/W	0h	High Speed Mode Enable Bits This bit determines if the High Speed mode is enabled. The correct GPIOs should be selected in the GPxGMUX/GPxMUX registers. Reset type: SYSRSn 0h (R/W) = SPI High Speed mode disabled. This is the default value after reset. 1h (R/W) = SPI High Speed mode enabled,
4	SPILBK	R/W	0h	SPI Loopback Mode Select Loopback mode allows module validation during device testing. This mode is valid only in CONTROLLER mode of the SPI. Reset type: SYSRSn 0h (R/W) = SPI loopback mode disabled. This is the default value after reset. 1h (R/W) = SPI loopback mode enabled, PICO/POCI lines are connected internally. Used for module self-tests.
3-0	SPICHAR	R/W	0h	Character Length Control Bits These four bits determine the number of bits to be shifted in or SPI CHAR0 out as a single character during one shift sequence. SPICHAR = Word length - 1 Reset type: SYSRSn 0h (R/W) = 1-bit word 1h (R/W) = 2-bit word 7h (R/W) = 8-bit word Fh (R/W) = 16-bit word

19.6.2.2 SPICTL Register (Offset = 1h) [Reset = 0000h]

SPICTL is shown in Figure 19-15 and described in Table 19-10.

Return to the Summary Table.

SPICTL controls data transmission, the SPI's ability to generate interrupts, the SPICLK phase, and the operational mode (PERIPHERAL or CONTROLLER).

Figure 19-15. SPICTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			OVERRUNINTENA	CLK_PHASE	CONTROLLER_PERIPHERAL	TALK	SPIINTENA
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-10. SPICTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4	OVERRUNINTENA	R/W	0h	Overrun Interrupt Enable Overrun Interrupt Enable. Setting this bit causes an interrupt to be generated when the RECEIVER OVERRUN Flag bit (SPISTS.7) is set by hardware. Interrupts generated by the RECEIVER OVERRUN Flag bit and the SPI INT FLAG bit (SPISTS.6) share the same interrupt vector. Reset type: SYSRSn 0h (R/W) = Disable RECEIVER OVERRUN interrupts. 1h (R/W) = Enable RECEIVER_OVERRUN interrupts.
3	CLK_PHASE	R/W	0h	SPI Clock Phase Select This bit controls the phase of the SPICLK signal. CLOCK PHASE and CLOCK POLARITY (SPICCR.6) make four different clocking schemes possible (see clocking figures in SPI chapter). When operating with CLOCK PHASE high, the SPI (CONTROLLER or PERIPHERAL) makes the first bit of data available after SPIDAT is written and before the first edge of the SPICLK signal, regardless of which SPI mode is being used. Reset type: SYSRSn 0h (R/W) = Normal SPI clocking scheme, depending on the CLOCK POLARITY bit (SPICCR.6). 1h (R/W) = SPICLK signal delayed by one half-cycle. Polarity determined by the CLOCK POLARITY bit.
2	CONTROLLER_PERIPHERAL	R/W	0h	SPI Network Mode Control This bit determines whether the SPI is a network CONTROLLER or PERIPHERAL. After SPI reset, SPI is automatically configured as a PERIPHERAL Reset type: SYSRSn 0h (R/W) = SPI is configured as a PERIPHERAL. 1h (R/W) = SPI is configured as a CONTROLLER.

Table 19-10. SPICTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	TALK	R/W	0h	Transmit Enable The TALK bit can disable data transmission (CONTROLLER or PERIPHERAL) by placing the serial data output in the high-impedance state. If this bit is disabled during a transmission, the transmit shift register continues to operate until the previous character is shifted out. When the TALK bit is disabled, the SPI is still able to receive characters and update the status flags. TALK is cleared (disabled) by a system reset. Reset type: SYSRSn 0h (R/W) = Disables transmission: - PERIPHERAL mode operation: If not previously configured as a general-purpose I/O pin, the SPIPOCI pin will be put in the high-impedance state. - CONTROLLER mode operation: If not previously configured as a general-purpose I/O pin, the SPIPICO pin will be put in the high-impedance state. 1h (R/W) = Enables transmission For the 4-pin option, ensure to enable the receiver's SPIPTEn input pin.
0	SPIINTENA	R/W	0h	SPI Interrupt Enable This bit controls the SPI's ability to generate a transmit/receive interrupt. The SPI INT FLAG bit (SPISTS.6) is unaffected by this bit. Reset type: SYSRSn 0h (R/W) = Disables the interrupt. 1h (R/W) = Enables the interrupt.

19.6.2.3 SPISTS Register (Offset = 2h) [Reset = 0000h]

SPISTS is shown in Figure 19-16 and described in Table 19-11.

Return to the Summary Table.

SPISTS contrains interrupt and status bits.

Figure 19-16. SPISTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
OVERRUN_FLAG	INT_FLAG	BUFFULL_FLAG	RESERVED				
W1C-0h	RC-0h	R-0h	R-0h				

Table 19-11. SPISTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	OVERRUN_FLAG	W1C	0h	SPI Receiver Overrun Flag This bit is a read/clear-only flag. The SPI hardware sets this bit when a receive or transmit operation completes before the previous character has been read from the buffer. The bit is cleared in one of three ways: - Writing a 1 to this bit - Writing a 0 to SPI SW RESET (SPICCR.7) - Resetting the system If the OVERRUN INT ENA bit (SPICTL.4) is set, the SPI requests only one interrupt upon the first occurrence of setting the RECEIVER OVERRUN Flag bit. Subsequent overruns will not request additional interrupts if this flag bit is already set. This means that in order to allow new overrun interrupt requests the user must clear this flag bit by writing a 1 to SPISTS.7 each time an overrun condition occurs. In other words, if the RECEIVER OVERRUN Flag bit is left set (not cleared) by the interrupt service routine, another overrun interrupt will not be immediately re-entered when the interrupt service routine is exited. Reset type: SYSRSn 0h (R/W) = A receive overrun condition has not occurred. 1h (R/W) = The last received character has been overwritten and therefore lost (when the SPIRXBUF was overwritten by the SPI module before the previous character was read by the user application). Writing a '1' will clear this bit. The RECEIVER OVERRUN Flag bit should be cleared during the interrupt service routine because the RECEIVER OVERRUN Flag bit and SPI INT FLAG bit (SPISTS.6) share the same interrupt vector. This will alleviate any possible doubt as to the source of the interrupt when the next byte is received.

Table 19-11. SPISTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	INT_FLAG	RC	0h	SPI Interrupt Flag SPI INT FLAG is a read-only flag. Hardware sets this bit to indicate that the SPI has completed sending or receiving the last bit and is ready to be serviced. This flag causes an interrupt to be requested if the SPI INT ENA bit (SPICTL.0) is set. The received character is placed in the receiver buffer at the same time this bit is set. This bit is cleared in one of three ways: - Reading SPIRXBUF - Writing a 0 to SPI SW RESET (SPICCR.7) - Resetting the system Note: This bit should not be used if FIFO mode is enabled. The internal process of copying the received word from SPIRXBUF to the Receive FIFO will clear this bit. Use the FIFO status, or FIFO interrupt bits for similar functionality. Reset type: SYSRSn 0h (R/W) = No full words have been received or transmitted. 1h (R/W) = Indicates that the SPI has completed sending or receiving the last bit and is ready to be serviced.
5	BUFFULL_FLAG	R	0h	SPI Transmit Buffer Full Flag This read-only bit gets set to 1 when a character is written to the SPI Transmit buffer SPITXBUF. It is cleared when the character is automatically loaded into SPIDAT when the shifting out of a previous character is complete. Reset type: SYSRSn 0h (R/W) = Transmit buffer is not full. 1h (R/W) = Transmit buffer is full.
4-0	RESERVED	R	0h	Reserved

19.6.2.4 SPIBRR Register (Offset = 4h) [Reset = 0000h]

SPIBRR is shown in Figure 19-17 and described in Table 19-12.

Return to the Summary Table.

SPIBRR contains the bits used for baud-rate selection.

Figure 19-17. SPIBRR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	SPI_BIT_RATE						
R-0h	R/W-0h						

Table 19-12. SPIBRR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-0	SPI_BIT_RATE	R/W	0h	SPI Baud Rate Control These bits determine the bit transfer rate if the SPI is the network SPI BIT RATE 0 CONTROLLER. There are 125 data-transfer rates (each a function of the CPU clock, LSPCLK) that can be selected. One data bit is shifted per SPICLK cycle. (SPICLK is the baud rate clock output on the SPICLK pin.) If the SPI is a network PERIPHERAL, the module receives a clock on the SPICLK pin from the network CONTROLLER. Therefore, these bits have no effect on the SPICLK signal. The frequency of the input clock from the CONTROLLER should not exceed the PERIPHERAL SPI's LSPCLK signal divided by 4. In CONTROLLER mode, the SPI clock is generated by the SPI and is output on the SPICLK pin. The SPI baud rates are determined by the following formula: For SPIBRR = 3 to 127: SPI Baud Rate = LSPCLK / (SPIBRR + 1) For SPIBRR = 0, 1, or 2: SPI Baud Rate = LSPCLK / 4 Reset type: SYSRSn 3h (R/W) = SPI Baud Rate = LSPCLK/4 4h (R/W) = SPI Baud Rate = LSPCLK/5 7Eh (R/W) = SPI Baud Rate = LSPCLK/127 7Fh (R/W) = SPI Baud Rate = LSPCLK/128

19.6.2.5 SPIRXEMU Register (Offset = 6h) [Reset = 0000h]

SPIRXEMU is shown in Figure 19-18 and described in Table 19-13.

Return to the Summary Table.

SPIRXEMU contains the received data. Reading SPIRXEMU does not clear the SPI INT FLAG bit of SPISTS. This is not a real register but a dummy address from which the contents of SPIRXBUF can be read by the emulator without clearing the SPI INT FLAG.

Figure 19-18. SPIRXEMU Register

15	14	13	12	11	10	9	8
ERXBn							
R-0h							
7	6	5	4	3	2	1	0
ERXBn							
R-0h							

Table 19-13. SPIRXEMU Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	ERXBn	R	0h	Emulation Buffer Received Data SPIRXEMU functions almost identically to SPIRXBUF, except that reading SPIRXEMU does not clear the SPI INT FLAG bit (SPISTS.6). Once the SPIDAT has received the complete character, the character is transferred to SPIRXEMU and SPIRXBUF, where it can be read. At the same time, SPI INT FLAG is set. This mirror register was created to support emulation. Reading SPIRXBUF clears the SPI INT FLAG bit (SPISTS.6). In the normal operation of the emulator, the control registers are read to continually update the contents of these registers on the display screen. SPIRXEMU was created so that the emulator can read this register and properly update the contents on the display screen. Reading SPIRXEMU does not clear the SPI INT FLAG bit, but reading SPIRXBUF clears this flag. In other words, SPIRXEMU enables the emulator to emulate the true operation of the SPI more accurately. It is recommended that you view SPIRXEMU in the normal emulator run mode. Reset type: SYSRSn

19.6.2.6 SPIRXBUF Register (Offset = 7h) [Reset = 0000h]

SPIRXBUF is shown in Figure 19-19 and described in Table 19-14.

Return to the Summary Table.

SPIRXBUF contains the received data. Reading SPIRXBUF clears the SPI INT FLAG bit in SPISTS. If FIFO mode is enabled, reading this register will also decrement the RXFFST counter in SPIFFRX.

Figure 19-19. SPIRXBUF Register

15	14	13	12	11	10	9	8
RXBn							
R-0h							
7	6	5	4	3	2	1	0
RXBn							
R-0h							

Table 19-14. SPIRXBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RXBn	R	0h	Received Data Once SPIDAT has received the complete character, the character is transferred to SPIRXBUF, where it can be read. At the same time, the SPI INT FLAG bit (SPISTS.6) is set. Since data is shifted into the SPI's most significant bit first, it is stored right-justified in this register. Reset type: SYSRSn

19.6.2.7 SPITXBUF Register (Offset = 8h) [Reset = 0000h]

SPITXBUF is shown in Figure 19-20 and described in Table 19-15.

Return to the Summary Table.

SPITXBUF stores the next character to be tranmitted. Writing to this register sets the TX BUF FULL Flag bit in SPISTS. When the transmission of the current character is complete, the contents of this register are automatically loaded in SPIDAT and the TX BUF FULL Flag is cleared. If no tranmission is currently active, data written to this register falls through into the SPIDAT register and the TX BUF FULL Flag is not set.
In CONTROLLER mode, if no tranmission is currently active, writing to this register initiates a transmission in the same manner that writing to SPIDAT does.

Figure 19-20. SPITXBUF Register

15	14	13	12	11	10	9	8
TXBn							
R/W-0h							
7	6	5	4	3	2	1	0
TXBn							
R/W-0h							

Table 19-15. SPITXBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	TXBn	R/W	0h	Transmit Data Buffer This is where the next character to be transmitted is stored. When the transmission of the current character has completed, if the TX BUF FULL Flag bit is set, the contents of this register is automatically transferred to SPIDAT, and the TX BUF FULL Flag is cleared. Writes to SPITXBUF must be left-justified. Reset type: SYSRSn

19.6.2.8 SPIDAT Register (Offset = 9h) [Reset = 0000h]

SPIDAT is shown in Figure 19-21 and described in Table 19-16.

Return to the Summary Table.

SPIDAT is the transmit and receive shift register. Data written to SPIDAT is shifted out (MSB) on subsequent SPICLK cycles. For every bit (MSB) shifted out of the SPI, a bit is shifted into the LSB end of the shift register.

Figure 19-21. SPIDAT Register

15	14	13	12	11	10	9	8
SDATn							
R/W-0h							
7	6	5	4	3	2	1	0
SDATn							
R/W-0h							

Table 19-16. SPIDAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	SDATn	R/W	0h	Serial Data Shift Register - It provides data to be output on the serial output pin if the TALK bit (SPICTL.1) is set. - When the SPI is operating as a CONTROLLER, a data transfer is initiated. When initiating a transfer, check the CLOCK POLARITY bit (SPICCR.6) described in Section 10.2.1.1 and the CLOCK PHASE bit (SPICTL.3) described in Section 10.2.1.2, for the requirements. In CONTROLLER mode, writing dummy data to SPIDAT initiates a receiver sequence. Since the data is not hardware-justified for characters shorter than sixteen bits, transmit data must be written in left-justified form, and received data read in right-justified form. Reset type: SYSRSn

19.6.2.9 SPIFFTX Register (Offset = Ah) [Reset = A000h]

SPIFFTX is shown in Figure 19-22 and described in Table 19-17.

Return to the Summary Table.

SPIFFTX contains both control and status bits related to the output FIFO buffer. This includes FIFO reset control, FIFO interrupt level control, FIFO level status, as well as FIFO interrupt enable and clear bits.

Figure 19-22. SPIFFTX Register

15	14	13	12	11	10	9	8
SPIRST	SPIFFENA	TXFIFO	TXFFST				
R/W-1h	R/W-0h	R/W-1h	R-0h				
7	6	5	4	3	2	1	0
TXFFINT	TXFFINTCLR	TXFFIENA	TXFFIL				
R-0h	W-0h	R/W-0h	R/W-0h				

Table 19-17. SPIFFTX Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SPIRST	R/W	1h	SPI Reset Reset type: SYSRSn 0h (R/W) = Write 0 to reset the SPI transmit and receive channels. The SPI FIFO register configuration bits will be left as is. 1h (R/W) = SPI FIFO can resume transmit or receive. No effect to the SPI registers bits.
14	SPIFFENA	R/W	0h	SPI FIFO Enhancements Enable Reset type: SYSRSn 0h (R/W) = SPI FIFO enhancements are disabled. 1h (R/W) = SPI FIFO enhancements are enabled.
13	TXFIFO	R/W	1h	TX FIFO Reset Reset type: SYSRSn 0h (R/W) = Write 0 to reset the FIFO pointer to zero, and hold in reset. 1h (R/W) = Release transmit FIFO from reset.
12-8	TXFFST	R	0h	Transmit FIFO Status Reset type: SYSRSn 0h (R/W) = Transmit FIFO is empty. 1h (R/W) = Transmit FIFO has 1 word. 2h (R/W) = Transmit FIFO has 2 words. 10h (R/W) = Transmit FIFO has 16 words, which is the maximum. 1Fh (R/W) = Reserved.
7	TXFFINT	R	0h	TX FIFO Interrupt Flag Reset type: SYSRSn 0h (R/W) = TXFIFO interrupt has not occurred, This is a read-only bit. 1h (R/W) = TXFIFO interrupt has occurred, This is a read-only bit.
6	TXFFINTCLR	W	0h	TXFIFO Interrupt Clear Reset type: SYSRSn 0h (R/W) = Write 0 has no effect on TXFIFINT flag bit, Bit reads back a zero. 1h (R/W) = Write 1 to clear SPIFFTX[TXFFINT] flag.
5	TXFFIENA	R/W	0h	TX FIFO Interrupt Enable Reset type: SYSRSn 0h (R/W) = TX FIFO interrupt based on TXFFIL match (less than or equal to) will be disabled. 1h (R/W) = TX FIFO interrupt based on TXFFIL match (less than or equal to) will be enabled.

Table 19-17. SPIFFTX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	TXFFIL	R/W	0h	Transmit FIFO Interrupt Level Bits Transmit FIFO will generate interrupt when the FIFO status bits (TXFFST4-0) and FIFO level bits (TXFFIL4-0) match (less than or equal to). Reset type: SYSRSn 0h (R/W) = A TX FIFO interrupt request is generated when there are no words remaining in the TX buffer. 1h (R/W) = A TX FIFO interrupt request is generated when there is 1 word or no words remaining in the TX buffer. 2h (R/W) = A TX FIFO interrupt request is generated when there is 2 words or fewer remaining in the TX buffer. 10h (R/W) = A TX FIFO interrupt request is generated when there are 16 words or fewer remaining in the TX buffer. 1Fh (R/W) = Reserved.

19.6.2.10 SPIFFRX Register (Offset = Bh) [Reset = 201Fh]

SPIFFRX is shown in Figure 19-23 and described in Table 19-18.

Return to the Summary Table.

SPIFFRX contains both control and status bits related to the input FIFO buffer. This includes FIFO reset control, FIFO interrupt level control, FIFO level status, as well as FIFO interrupt enable and clear bits.

Figure 19-23. SPIFFRX Register

15	14	13	12	11	10	9	8
RXFFOVF	RXFFOVFCLR	RXFIFORESET	RXFFST				
R-0h	W-0h	R/W-1h	R-0h				
7	6	5	4	3	2	1	0
RXFFINT	RXFFINTCLR	RXFFIENA	RXFFIL				
R-0h	W-0h	R/W-0h	R/W-1Fh				

Table 19-18. SPIFFRX Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RXFFOVF	R	0h	Receive FIFO Overflow Flag Reset type: SYSRSn 0h (R/W) = Receive FIFO has not overflowed. This is a read-only bit. 1h (R/W) = Receive FIFO has overflowed, read-only bit. More than 16 words have been received in to the FIFO, and the first received word is lost.
14	RXFFOVFCLR	W	0h	Receive FIFO Overflow Clear Reset type: SYSRSn 0h (R/W) = Write 0 does not affect RXFFOVF flag bit, Bit reads back a zero. 1h (R/W) = Write 1 to clear SPIFFRX[RXFFOVF].
13	RXFIFORESET	R/W	1h	Receive FIFO Reset Reset type: SYSRSn 0h (R/W) = Write 0 to reset the FIFO pointer to zero, and hold in reset. 1h (R/W) = Re-enable receive FIFO operation.
12-8	RXFFST	R	0h	Receive FIFO Status Reset type: SYSRSn 0h (R/W) = Receive FIFO is empty. 1h (R/W) = Receive FIFO has 1 word. 2h (R/W) = Receive FIFO has 2 words. 10h (R/W) = Receive FIFO has 16 words, which is the maximum. 1Fh (R/W) = Reserved.
7	RXFFINT	R	0h	Receive FIFO Interrupt Flag Reset type: SYSRSn 0h (R/W) = RXFIFO interrupt has not occurred. This is a read-only bit. 1h (R/W) = RXFIFO interrupt has occurred. This is a read-only bit.
6	RXFFINTCLR	W	0h	Receive FIFO Interrupt Clear Reset type: SYSRSn 0h (R/W) = Write 0 has no effect on RXFIFINT flag bit, Bit reads back a zero. 1h (R/W) = Write 1 to clear SPIFFRX[RXFFINT] flag
5	RXFFIENA	R/W	0h	RX FIFO Interrupt Enable Reset type: SYSRSn 0h (R/W) = RX FIFO interrupt based on RXFFIL match (greater than or equal to) will be disabled. 1h (R/W) = RX FIFO interrupt based on RXFFIL match (greater than or equal to) will be enabled.

Table 19-18. SPIFFRX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	RXFFIL	R/W	1Fh	Receive FIFO Interrupt Level Bits Receive FIFO generates an interrupt when the FIFO status bits (RXFFST4-0) are greater than or equal to the FIFO level bits (RXFFIL4-0). The default value of these bits after reset is 11111. This avoids frequent interrupts after reset, as the receive FIFO will be empty most of the time. Reset type: SYSRSn 0h (R/W) = A RX FIFO interrupt request is generated when there is 0 or more words in the RX buffer. 1h (R/W) = A RX FIFO interrupt request is generated when there are 1 or more words in the RX buffer. 2h (R/W) = A RX FIFO interrupt request is generated when there are 2 or more words in the RX buffer. 10h (R/W) = A RX FIFO interrupt request is generated when there are 16 words in the RX buffer. 1Fh (R/W) = Reserved.

19.6.2.11 SPIFFCT Register (Offset = Ch) [Reset = 0000h]

SPIFFCT is shown in Figure 19-24 and described in Table 19-19.

Return to the Summary Table.

SPIFFCT controls the FIFO transmit delay bits.

Figure 19-24. SPIFFCT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TXDLY							
R/W-0h							

Table 19-19. SPIFFCT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	TXDLY	R/W	0h	FIFO Transmit Delay Bits These bits define the delay between every transfer from FIFO transmit buffer to transmit shift register. The delay is defined in number SPI serial clock cycles. The 8-bit register could define a minimum delay of 0 serial clock cycles and a maximum of 255 serial clock cycles. In FIFO mode, the buffer (TXBUF) between the shift register and the FIFO should be filled only after the shift register has completed shifting of the last bit. This is required to pass on the delay between transfers to the data stream. In the FIFO mode TXBUF should not be treated as one additional level of buffer. Reset type: SYSRSn 0h (R/W) = The next word in the TX FIFO buffer is transferred to SPITXBUF immediately upon completion of transmission of the previous word. 1h (R/W) = The next word in the TX FIFO buffer is transferred to SPITXBUF1 serial clock cycle after completion of transmission of the previous word. 2h (R/W) = The next word in the TX FIFO buffer is transferred to SPITXBUF 2 serial clock cycles after completion of transmission of the previous word. FFh (R/W) = The next word in the TX FIFO buffer is transferred to SPITXBUF 255 serial clock cycles after completion of transmission of the previous word.

19.6.2.12 SPIPRI Register (Offset = Fh) [Reset = 0000h]

SPIPRI is shown in Figure 19-25 and described in Table 19-20.

Return to the Summary Table.

SPIPRI controls auxillary functions for the SPI including emulation control, SPIPTE inversion, and 3-wire control.

Figure 19-25. SPIPRI Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	SOFT	FREE	RESERVED		PTEINV	TRIWIRE
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h

Table 19-20. SPIPRI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	SOFT	R/W	0h	Emulation Soft Run This bit only has an effect when the FREE bit is 0. Reset type: SYSRSn 0h (R/W) = Transmission stops midway in the bit stream while TSUSPEND is asserted. Once TSUSPEND is deasserted without a system reset, the remainder of the bits pending in the DATBUF are shifted. Example: If SPIDAT has shifted 3 out of 8 bits, the communication freezes right there. However, if TSUSPEND is later deasserted without resetting the SPI, SPI starts transmitting from where it had stopped (fourth bit in this case) and will transmit 8 bits from that point. 1h (R/W) = If the emulation suspend occurs before the start of a transmission, (that is, before the first SPICLK pulse) then the transmission will not occur. If the emulation suspend occurs after the start of a transmission, then the data will be shifted out to completion. When the start of transmission occurs is dependent on the baud rate used. Standard SPI mode: Stop after transmitting the words in the shift register and buffer. That is, after TXBUF and SPIDAT are empty. In FIFO mode: Stop after transmitting the words in the shift register and buffer. That is, after TX FIFO and SPIDAT are empty.
4	FREE	R/W	0h	Emulation Free Run These bits determine what occurs when an emulation suspend occurs (for example, when the debugger hits a breakpoint). The peripheral can continue whatever it is doing (free-run mode) or, if in stop mode, it can either stop immediately or stop when the current operation (the current receive/transmit sequence) is complete. Reset type: SYSRSn 0h (R/W) = Emulation mode is selected by the SOFT bit 1h (R/W) = Free run, continue SPI operation regardless of suspend or when the suspend occurred.
3-2	RESERVED	R	0h	Reserved

Table 19-20. SPIPRI Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PTEINV	R/W	0h	SPIPTEn Inversion Bit On devices with 2 or more SPI modules, inverting the SPIPTE signal on one of the modules allows the device to receive left and right-channel digital audio data. This bit is only applicable to PERIPHERAL mode. Writing to this bit while configured as CONTROLLER (CONTROLLER_PERIPHERAL = 1) has no effect Reset type: SYSRSn 0h (R/W) = SPIPTEn is active low (normal) 1h (R/W) = SPIPTE is active high (inverted)
0	TRIWIRE	R/W	0h	SPI 3-wire Mode Enable Reset type: SYSRSn 0h (R/W) = Normal 4-wire SPI mode. 1h (R/W) = 3-wire SPI mode enabled. The unused pin becomes a GPIO pin. In CONTROLLER mode, the SPIPICO pin becomes the SPICOCI (CONTROLLER receive and transmit) pin and SPIPOCI is free for non-SPI use. In PERIPHERAL mode, the SPIPOCI pin becomes the SPIPIPO (PERIPHERAL receive and transmit) pin and SPIPICO is free for non-SPI use.

Chapter 20

Inter-Integrated Circuit Module (I2C)

This chapter describes the features and operation of the inter-integrated circuit (I2C) module. The I2C module provides an interface between one of these devices and devices compliant with the NXP Semiconductors Inter-IC bus (I2C bus) specification version 2.1, and connected by way of an I2C bus. External components attached to this 2-wire serial bus can transmit/receive 1- to 8-bit data to/from the device through the I2C module. This chapter assumes the reader is familiar with the I2C bus specification.

Note

A unit of data transmitted or received by the I2C module can have fewer than 8 bits; however, for convenience, a unit of data is called a data byte throughout this chapter. The number of bits in a data byte is selectable by way of the BC bits of the mode register, I2CMDR.

20.1 Introduction

The I2C module supports any target or controller I2C-compatible device. Figure 20-1 shows an example of multiple I2C modules connected for a two-way transfer from one device to other devices.

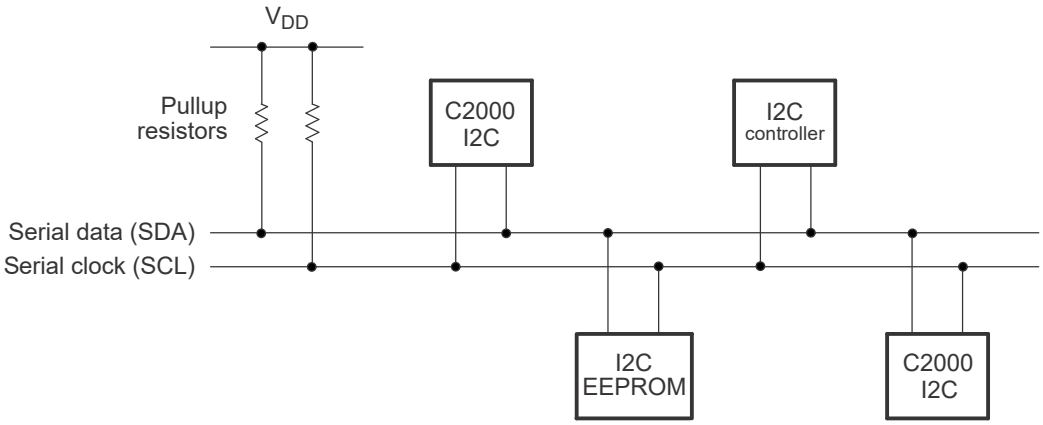


Figure 20-1. Multiple I2C Modules Connected

20.1.1 I2C Related Collateral

Foundational Materials

- C28x Academy - I2C
- I2C Hardware Overview (Video)
- I2C Protocol Overview (Video)
- *Understanding the I2C Bus Application Report*

Getting Started Materials

- *Configuring the TMS320F280x DSP as an I2C Processor Application Report*
- I2C Buffers Overview (Video)
- *I2C Dynamic Addressing Application Report*
- I2C translators overview (Video)
- *Interfacing EEPROM Using C2000 I2C Module Application Report*
- *Why, When, and How to use I2C Buffers Application Report*

Expert Materials

- *I2C Bus Pull-Up Resistor Calculation Application Report*
- *Maximum Clock Frequency of I2C Bus Using Repeaters Application Report*

20.1.2 Features

The I2C module has the following features:

- Compliance with the NXP Semiconductors I2C bus specification (version 2.1):
  - Support for 8-bit format transfers
  - 7-bit and 10-bit addressing modes
  - General call
  - START byte mode
  - Support for multiple controller-transmitters and target-receivers
  - Support for multiple target-transmitters and controller-receivers
  - Combined controller transmit/receive and receive/transmit mode
  - Data transfer rate from 10kbps up to 400kbps (Fast-mode)
- Extended Automatic Clock Stretching and Manual Clock Stretching modes
- Receive FIFO and Transmitter FIFO (16-deep x 8-bit FIFO)
- Supports two ePIE interrupts:
  - I2Cx Interrupt – Any of the following events can be configured to generate an I2Cx interrupt:
    - Transmit-data ready
    - Receive-data ready
    - Register-access ready
    - No-acknowledgment received
    - Arbitration lost
    - Stop condition detected
    - Addressed as target
    - Extended Automatic Clock Stretch
  - I2Cx_FIFO interrupts:
    - Transmit FIFO interrupt
    - Receive FIFO interrupt
- Module enable and disable capability
- Free data format mode

20.1.3 Features Not Supported

The I2C module does not support:

- High-speed mode (Hs-mode)
- CBUS-compatibility mode

20.1.4 Functional Overview

Each device connected to an I2C bus is recognized by a unique address. Each device can operate as either a transmitter or a receiver, depending on the function of the device. A device connected to the I2C bus can also be considered as the controller or the target when performing data transfers. A controller device is the device that initiates a data transfer on the bus and generates the clock signals to permit that transfer. During this transfer, any device addressed by this controller is considered a target. The I2C module supports the multi-controller mode, in which one or more devices capable of controlling an I2C bus can be connected to the same I2C bus.

For data communication, the I2C module has a serial data pin (SDA) and a serial clock pin (SCL), as shown in Figure 20-2. These two pins carry information between the C28x device and other devices connected to the I2C bus. The SDA and SCL pins are both bidirectional and each must be connected to a positive supply voltage using a pull-up resistor. When the bus is free, both pins are high. The driver of these two pins has an open-drain configuration to perform the required wired-AND function.

There are two major transfer techniques:

- Standard Mode: Send exactly n data values, where n is a value you program in an I2C module register. See the I2CCNT register in the *I2C Registers* section for more information.
- Repeat Mode: Keep sending data values until you use software to initiate a STOP condition or a new START condition. See the I2CMDR register in the *I2C Registers* section for RM bit information.

The I2C module consists of the following primary blocks:

- A serial interface: one data pin (SDA) and one clock pin (SCL)
- Data registers and FIFOs to temporarily hold receive data and transmit data traveling between the SDA pin and the CPU
- Control and status registers
- A peripheral bus interface to enable the CPU to access the I2C module registers and FIFOs.
- A clock synchronizer to synchronize the I2C input clock (from the device clock generator) and the clock on the SCL pin, and to synchronize data transfers with controllers of different clock speeds
- A prescaler to divide down the input clock that is driven to the I2C module
- A noise filter on each of the two pins, SDA and SCL
- An arbitrator to handle arbitration between the I2C module (when the I2C module is a controller) and another controller
- Interrupt generation logic, so that an interrupt can be sent to the CPU
- FIFO interrupt generation logic, so that FIFO access can be synchronized to data reception and data transmission in the I2C module

Figure 20-2 shows the four registers used for transmission and reception in non-FIFO mode. The CPU writes data for transmission to I2CDXR and reads received data from I2CDRR. When the I2C module is configured as a transmitter, data written to I2CDXR is copied to I2CXSR and shifted out on the SDA pin one bit at a time. When the I2C module is configured as a receiver, received data is shifted into I2CRSR and then copied to I2CDRR.

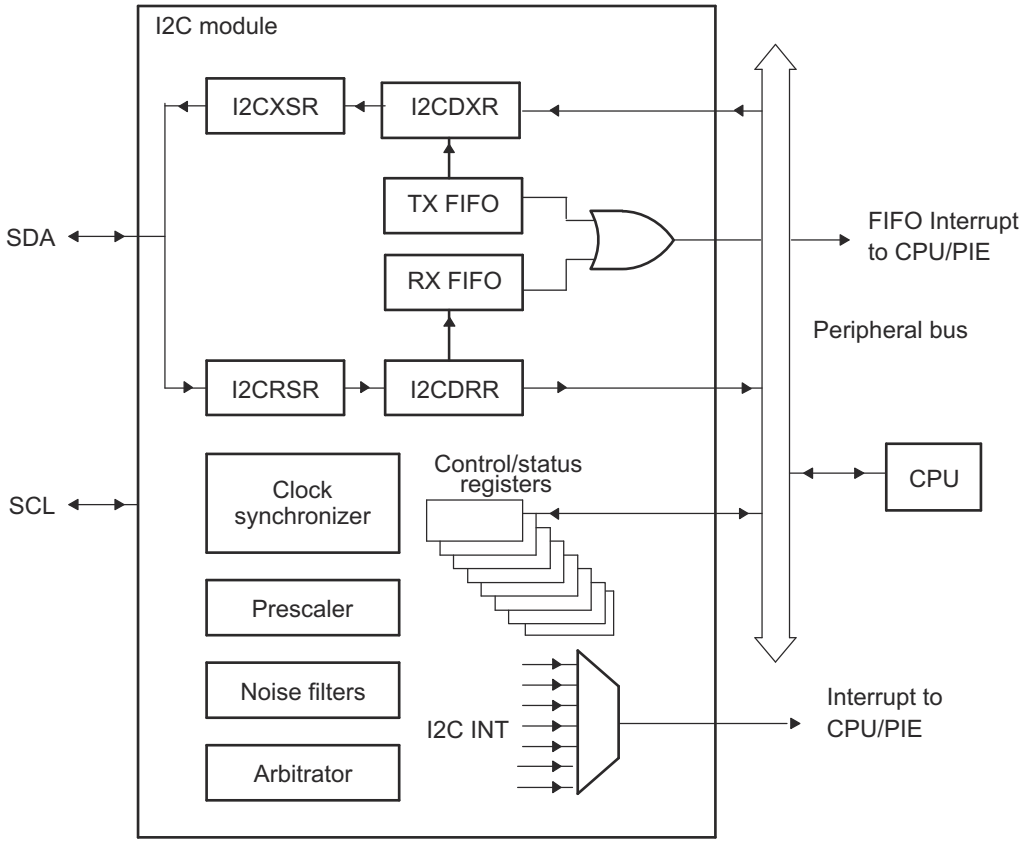


Figure 20-2. I2C Module Conceptual Block Diagram

20.1.5 Clock Generation

The I2C module clock determines the frequency at which the I2C module operates. A programmable prescaler in the I2C module divides down the SYSCLK to produce the I2C module clock and this I2C module clock is divided further to produce the I2C controller clock on the SCL pin. Figure 20-3 shows the clock generation diagram for I2C module.

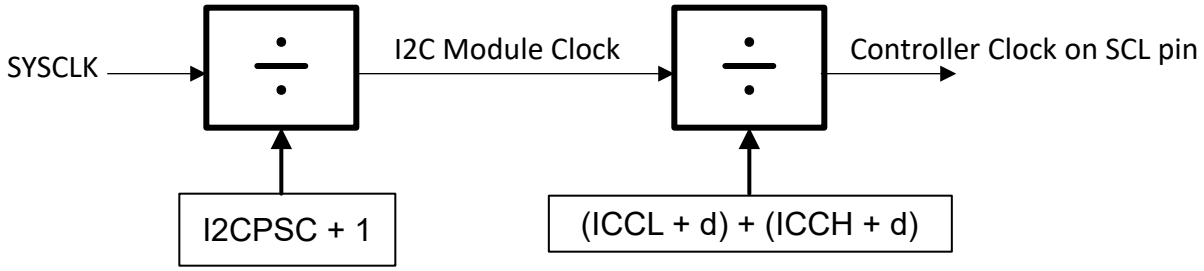


Figure 20-3. Clocking Diagram for the I2C Module

Note

To meet all of the I2C protocol timing specifications, the I2C module clock must be between 7 to 12MHz.

To specify the divide-down value, initialize the IPSC field of the prescaler register, I2CPSC. The resulting frequency is:

$$\text{I2C Module Clock (Fmod)} = \frac{\text{SYSCLK}}{(\text{I2CPSC} + 1)} \quad (25)$$

The prescaler must be initialized only while the I2C module is in the reset state (IRS = 0 in I2CMDR). The prescaled frequency takes effect only when IRS is changed to 1. Changing the IPSC value while IRS = 1 has no effect.

The controller clock appears on the SCL pin when the I2C module is configured to be a controller on the I2C bus. This clock controls the timing of communication between the I2C module and a target. As shown in Figure 20-3, a second clock divider in the I2C module divides down the module clock to produce the controller clock. The clock divider uses the ICCL value of I2CCLKL to divide down the low portion of the module clock signal and uses the ICCH value of I2CCLKH to divide down the high portion of the module clock signal. See Section 20.1.6 for the controller clock frequency equation.

20.1.6 I2C Clock Divider Registers (I2CCLKL and I2CCLKH)

As explained in Section 20.1.5, when the I2C module is a controller, the I2C module clock is divided down further to use as the controller clock on the SCL pin. As shown in Figure 20-4, the shape of the controller clock depends on two divide-down values:

- ICCL in I2CCLKL. For each controller clock cycle, ICCL determines the amount of time the signal is low.
- ICCH in I2CCKLH. For each controller clock cycle, ICCH determines the amount of time the signal is high.

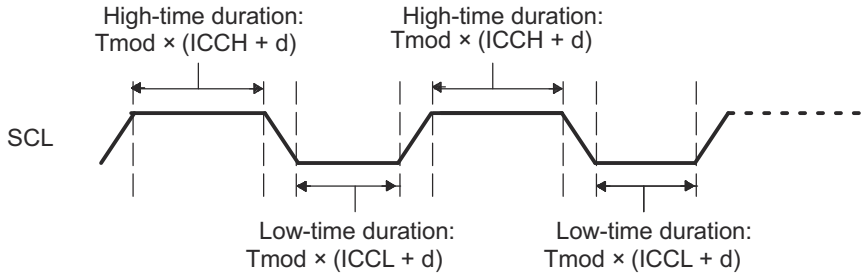


Figure 20-4. Roles of the Clock Divide-Down Values (ICCL and ICCH)

20.1.6.1 Formula for the Controller Clock Period

The controller clock period (Tmst) is a multiple of the period of the I2C Module Clock (Tmod):

$$\text{Controller Clock period (Tmst)} = \frac{[\,(\text{ICCH} + d) + (\text{ICCL} + d)\,]}{\text{I2C Module Clock (Fmod)}} \quad (26)$$

where d depends on the divide-down value IPSC, as shown in Table 20-1. IPSC is described in the I2CPSC register.

Table 20-1. Dependency of Delay d on the Divide-Down Value IPSC

IPSC	d
0	7
1	6
Greater than 1	5

20.2 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

Some IO functionality is defined by GPIO register settings independent of this peripheral. For input signals, the GPIO input qualification must be set to asynchronous mode by setting the appropriate GPxQSELn register bits to 11b. The internal pullups can be configured in the GPyPUD register.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

Note

To support a wider range of I2C IO levels, certain GPIOs have configurable fail-safe systems, V_{IH} minimum thresholds, and configurable sinking capabilities.

20.3 I2C Module Operational Details

This section provides an overview of the I2C bus protocol and how it is implemented.

20.3.1 Input and Output Voltage Levels

One clock pulse is generated by the controller device for each data bit transferred. Due to a variety of different technology devices that can be connected to the I2C bus, the levels of logic 0 (low) and logic 1 (high) are not fixed and depend on the associated level of V_{DD}. For details, see the device data sheet.

20.3.2 Selecting Pullup Resistors

The chosen pullup resistor must meet the I2C standard timings. In most circumstances, 2.2kΩ of total bus resistance to VDDIO is sufficient. The value of the pullup resistance used on both the SCL and SDA pins be matched is also recommended. For evaluating pullup resistor values for a particular design, see the *I2C Bus Pullup Resistor Calculation Application Report*.

20.3.3 Data Validity

The data on SDA must be stable during the high period of the clock (see Figure 20-5). The high or low state of the data line, SDA, must change only when the clock signal on SCL is low.

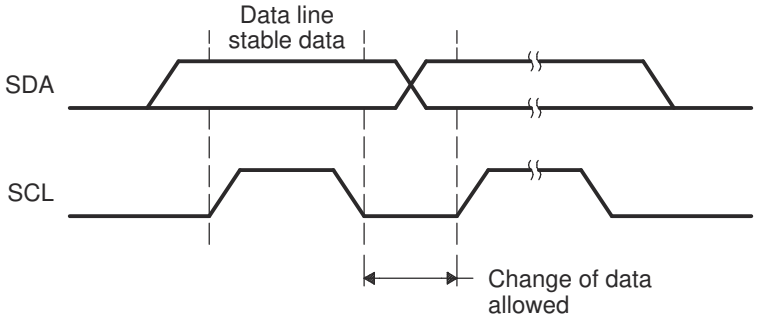


Figure 20-5. Bit Transfer on the I2C bus

20.3.4 Operating Modes

The I2C module has four basic operating modes to support data transfers as a controller and as a target. See Table 20-2 for the names and descriptions of the modes.

If the I2C module is a controller, the I2C module begins as a controller-transmitter and typically transmits an address for a particular target. When giving data to the target, the I2C module must remain a controller-transmitter. To receive data from a target, the I2C module must be changed to the controller-receiver mode.

If the I2C module is a target, the I2C module begins as a target-receiver and typically sends acknowledgment when the I2C module recognizes the target address from a controller. If the controller is sending data to the I2C module, the module must remain a target-receiver. If the controller has requested data from the I2C module, the module must be changed to the target-transmitter mode.

Table 20-2. Operating Modes of the I2C Module

Operating Mode	Description
Target-receiver mode	The I2C module is a target and receives data from a controller. All targets begin in this mode. In this mode, serial data bits received on SDA are shifted in with the clock pulses that are generated by the controller. As a target, the I2C module does not generate the clock signal, but can hold SCL low while the intervention of the device is required (RSFULL = 1 in I2CSTR) after a byte has been received. See Section 20.3.8 for more details.
Target-transmitter mode	The I2C module is a target and transmits data to a controller. This mode can be entered only from the target-receiver mode; the I2C module must first receive a command from the controller. When using any of the 7-bit/10-bit addressing formats, the I2C module enters the target-transmitter mode if the target address byte is the same as the address (in I2COAR) and the controller has transmitted R/ $\overline{W}$ = 1. As a target-transmitter, the I2C module then shifts the serial data out on SDA with the clock pulses that are generated by the controller. While a target, the I2C module does not generate the clock signal, but it can hold SCL low while the intervention of the device is required (XSMT = 0 in I2CSTR) after a byte has been transmitted. See Section 20.3.8 for more details.
Controller-receiver mode	The I2C module is a controller and receives data from a target. This mode can be entered only from the controller-transmitter mode; the I2C module must first transmit a command to the target. When using any of the 7-bit/10-bit addressing formats, the I2C module enters the controller-receiver mode after transmitting the target address byte and R/ $\overline{W}$ = 1. Serial data bits on SDA are shifted into the I2C module with the clock pulses generated by the I2C module on SCL. The clock pulses are inhibited and SCL is held low when the intervention of the device is required (RSFULL = 1 in I2CSTR) after a byte has been received.
Controller-transmitter mode	The I2C module is a controller and transmits control information and data to a target. All controllers begin in this mode. In this mode, data assembled in any of the 7-bit/10-bit addressing formats is shifted out on SDA. The bit shifting is synchronized with the clock pulses generated by the I2C module on SCL. The clock pulses are inhibited and SCL is held low when the intervention of the device is required (XSMT = 0 in I2CSTR) after a byte has been transmitted.

To summarize, SCL is held low in the following conditions:

- When an overrun condition is detected (RSFULL = 1), in Target-receiver mode.
- When an underflow condition is detected (XSMT = 0), in Target-transmitter mode.

I2C target nodes accept and provide data when the I2C controller node requests data.

- To release SCL in target-receiver mode, read data from I2CDRR.
- To release SCL in target-transmitter mode, write data to I2CDXR.
- To force a release without handling the data, reset the module using the I2CMDR.IRS bit.

Table 20-3. Controller-Transmitter/Receiver Bus Activity Defined by the RM, STT, and STP Bits of I2CMDR

RM	STT	STP	Bus Activity(1)	Description
0	0	0	None	No activity
0	0	1	P	STOP condition
0	1	0	S-A-D..(n)..D.	START condition, target address, n data bytes (n = value in I2CCNT)
0	1	1	S-A-D..(n)..D-P	START condition, target address, n data bytes, STOP condition (n = value in I2CCNT)
1	0	0	None	No activity
1	0	1	P	STOP condition
1	1	0	S-A-D-D-D.	Repeat mode transfer: START condition, target address, continuous data transfers until STOP condition or next START condition
1	1	1	None	Reserved bit combination (No activity)

(1) S = START condition; A = Address; D = Data byte; P = STOP condition;

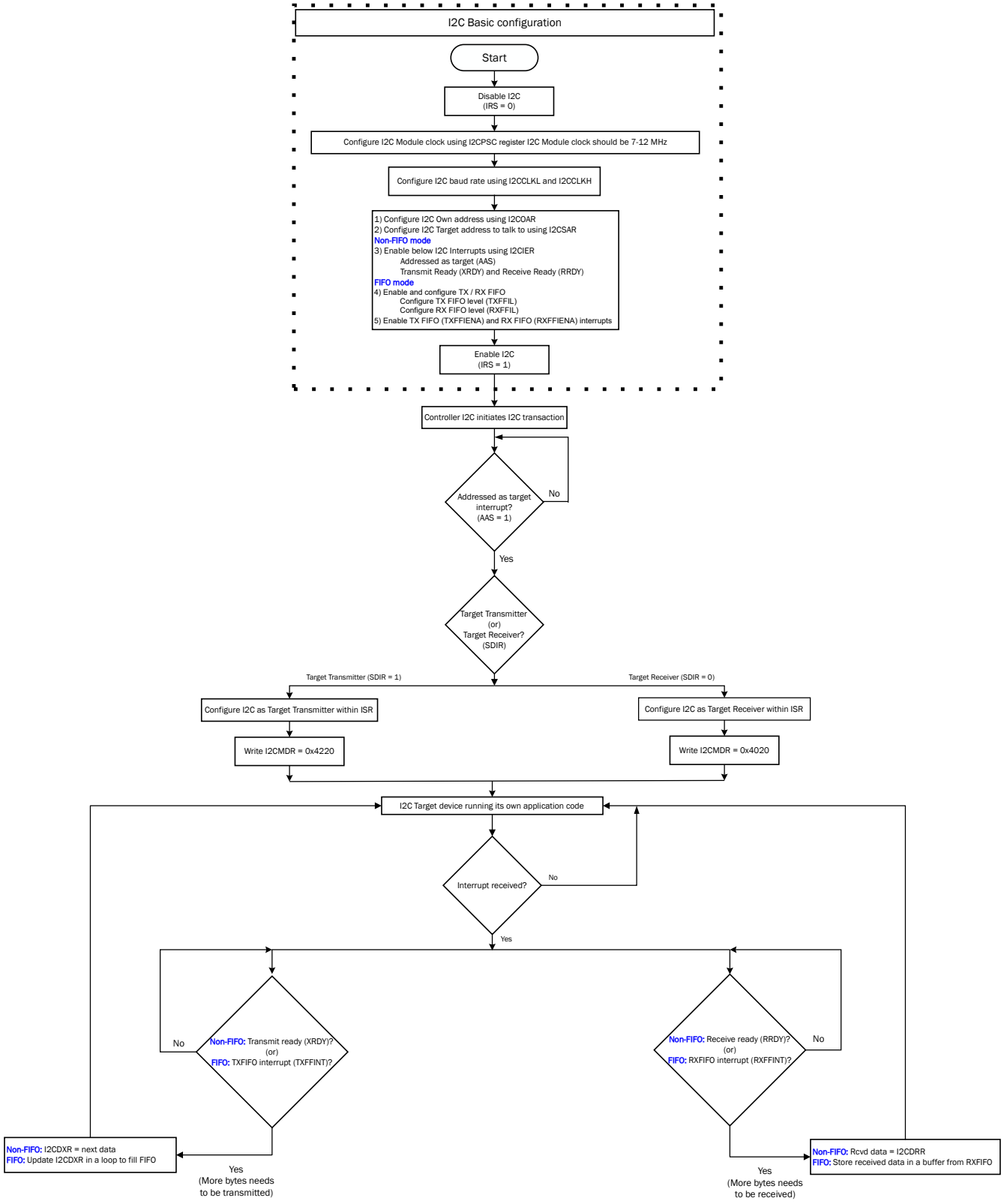


Figure 20-6. I2C Target TX / RX Flowchart

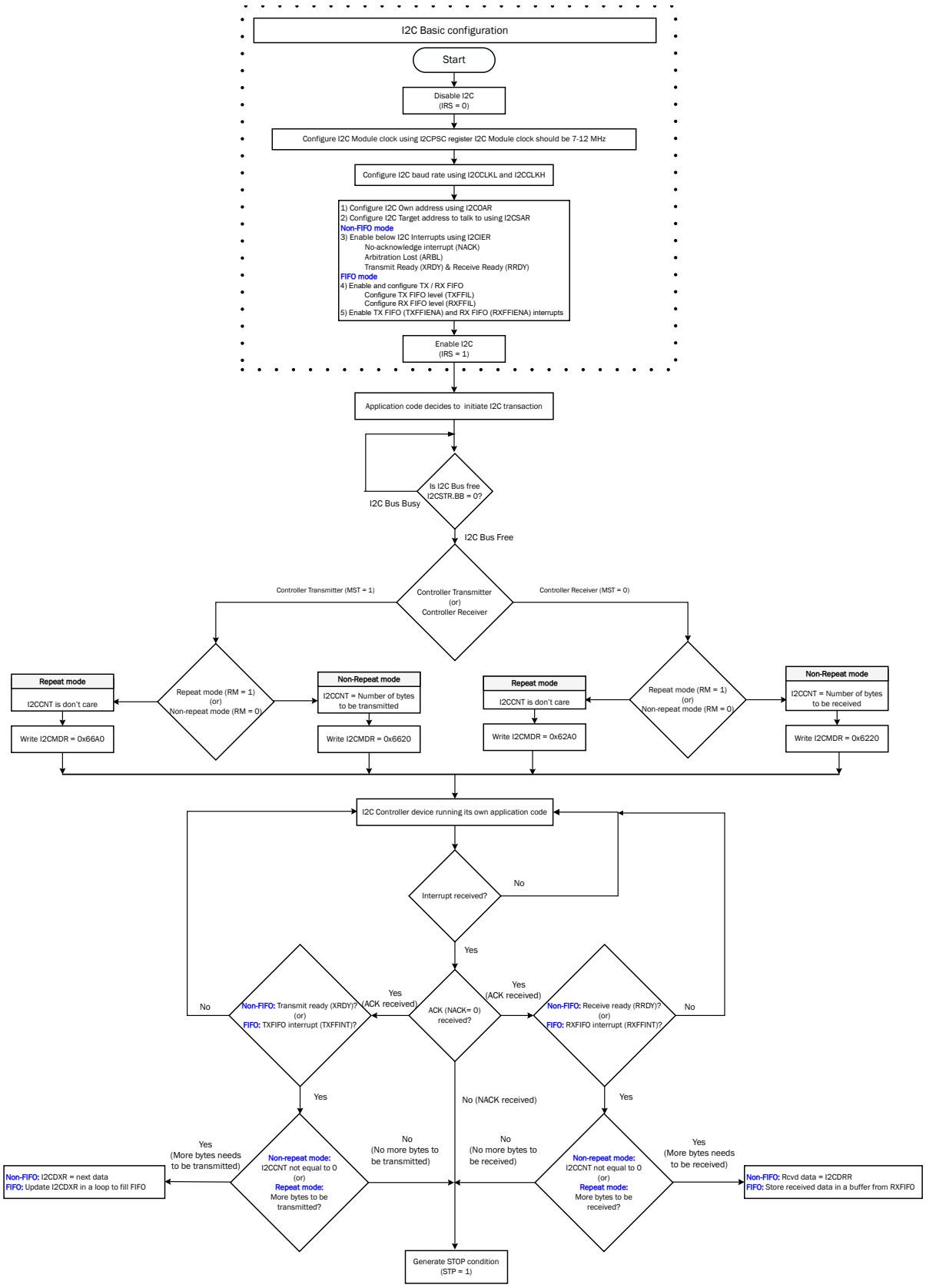


Figure 20-7. I2C Controller TX / RX Flowchart

20.3.5 I2C Module START and STOP Conditions

START and STOP conditions can be generated by the I2C module when the module is configured to be a controller on the I2C bus. As shown in Figure 20-8:

- The START condition is defined as a high-to-low transition on the SDA line while SCL is high. A controller drives this condition to indicate the start of a data transfer.
- The STOP condition is defined as a low-to-high transition on the SDA line while SCL is high. A controller drives this condition to indicate the end of a data transfer.

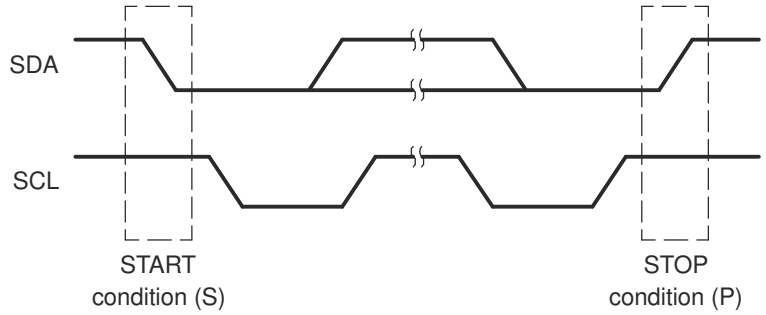


Figure 20-8. I2C Module START and STOP Conditions

After a START condition and before a subsequent STOP condition, the I2C bus is considered busy, and the bus busy (BB) bit of I2CSTR is 1. Between a STOP condition and the next START condition, the bus is considered free, and BB is 0.

For the I2C module to start a data transfer with a START condition, the controller mode bit (MST) and the START condition bit (STT) in I2CMDR must both be 1. For the I2C module to end a data transfer with a STOP condition, the STOP condition bit (STP) must be set to 1. When the BB bit is set to 1 and the STT bit is set to 1, a repeated START condition is generated. For a description of I2CMDR and the bits (including MST, STT, and STP), see Section 20.7.

The I2C peripheral cannot detect a START or STOP condition while in reset (IRS = 0). The BB bit remains in the cleared state (BB = 0) while the I2C peripheral is in reset (IRS = 0). When the I2C peripheral is taken out of reset (IRS set to 1), the BB bit does not correctly reflect the I2C bus status until a START or STOP condition is detected.

Follow these steps before initiating the first data transfer with I2C:

1. After taking the I2C peripheral out of reset by setting the IRS bit to 1, wait a period larger than the total time taken for the longest data transfer in the application. By waiting for a period of time after I2C comes out of reset, make sure that at least one START or STOP condition has occurred on the I2C bus and has been captured by the BB bit. After this period, the BB bit correctly reflects the state of the I2C bus.
2. Check the BB bit and verify that BB = 0 (bus not busy) before proceeding.
3. Begin data transfers.

Not resetting the I2C peripheral in between transfers makes sure that the BB bit reflects the actual bus status. If users must reset the I2C peripheral in between transfers, repeat steps 1 through 3 every time the I2C peripheral is taken out of reset.

20.3.6 Non-repeat Mode versus Repeat Mode

Non-repeat mode:

- When I2CMDR.RM = 0, I2C module is configured in non-repeat mode.
- I2CCNT register determines the number of bytes to be transmitted or received.
- If STP = 0 in I2CMDR, the ARDY bit is set when the internal data counter counts down to 0.
- If STP = 1, ARDY bit does not get set and I2C module generates a STOP condition when the internal data counter counts down to 0.

Note

In non-repeat mode (RM = 0), if I2CCNT is set to 0, I2C state machine expects to transmit or receive 65536 bytes and not 0 bytes.

Repeat mode:

- When I2CMDR.RM = 1, I2C module is configured in repeat mode.
- I2CCNT register contents do not determine the number of bytes to be transmitted or received.
- Number of bytes to be transmitted or received can be controlled by software.
- ARDY bit gets set at end of transmission and reception of each byte.

Note

Once you start I2C transaction in non-repeat mode or repeat mode, you cannot switch into another mode until the I2C transaction is completed with a STOP condition.

20.3.7 Serial Data Formats

Figure 20-9 shows an example of a data transfer on the I2C bus. The I2C module supports 1 to 8-bit data values. In Figure 20-9, 8-bit data is transferred. Each bit put on the SDA line equates to 1 pulse on the SCL line, and the values are always transferred with the most significant bit (MSB) first. The number of data values that can be transmitted or received is unrestricted. The serial data format used in Figure 20-9 is the 7-bit addressing format. The I2C module supports the formats shown in Figure 20-10 through Figure 20-12 and described in the paragraphs that follow the figures.

Note

In Figure 20-9 through Figure 20-12, n = the number of data bits (from 1 to 8) specified by the bit count (BC) field of I2CMDR.

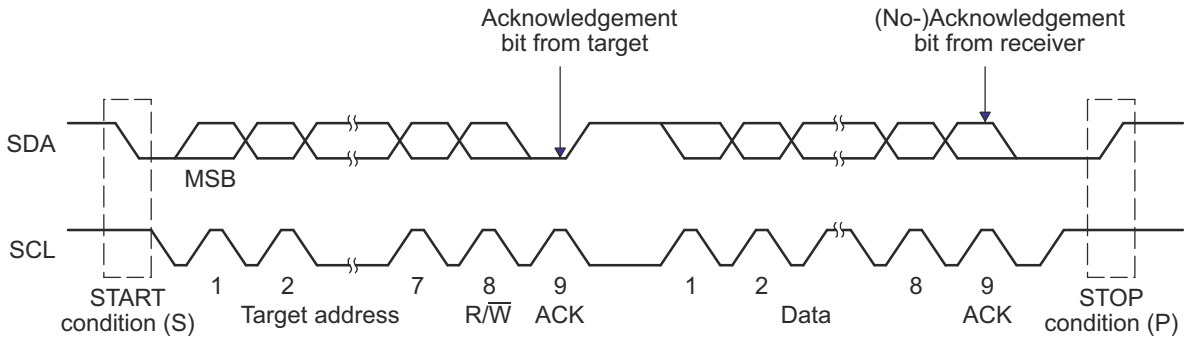


Figure 20-9. I2C Module Data Transfer (7-Bit Addressing with 8-bit Data Configuration Shown)

20.3.7.1 7-Bit Addressing Format

The 7-bit addressing format is the default format after reset. Disabling expanded address (I2CMDR.XA = 0) and free data format (I2CMDR.FDF = 0) enables 7-bit addressing format.

In this format (see Figure 20-10), the first byte after a START condition (S) consists of a 7-bit target address followed by a R/$\overline{W}$ bit. R/$\overline{W}$ determines the direction of the data:

- R/$\overline{W}$ = 0: The I2C controller writes (transmits) data to the addressed target. This can be achieved by setting I2CMDR.TRX = 1 (Transmitter mode)
- R/$\overline{W}$ = 1: The I2C controller reads (receives) data from the target. This can be achieved by setting I2CMDR.TRX = 0 (Receiver mode)

An extra clock cycle dedicated for acknowledgment (ACK) is inserted after each byte. If the ACK bit is inserted by the target after the first byte from the controller, it is followed by n bits of data from the transmitter (controller or target, depending on the R/$\overline{W}$ bit). n is a number from 1 to 8 determined by the bit count (BC) field of I2CMDR. After the data bits have been transferred, the receiver inserts an ACK bit.

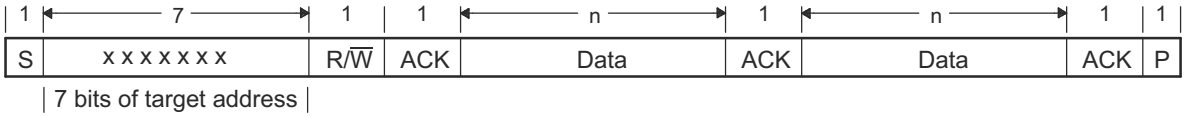


Figure 20-10. I2C Module 7-Bit Addressing Format (FDF = 0, XA = 0 in I2CMDR)

20.3.7.2 10-Bit Addressing Format

The 10-bit addressing format can be enabled by setting expanded address (I2CMDR.XA = 1) and disabling free data format (I2CMDR.FDF = 0).

The 10-bit addressing format (see Figure 20-11) is similar to the 7-bit addressing format, but the controller sends the target address in two separate byte transfers. The first byte consists of 11110b, the two MSBs of the 10-bit target address, and R/$\overline{W}$. The second byte is the remaining 8 bits of the 10-bit target address. The target must send acknowledgment after each of the two byte transfers. Once the controller has written the second byte to the target, the controller can either write data or use a repeated START condition to change the data direction. For more details about using 10-bit addressing, see the NXP Semiconductors I2C bus specification.

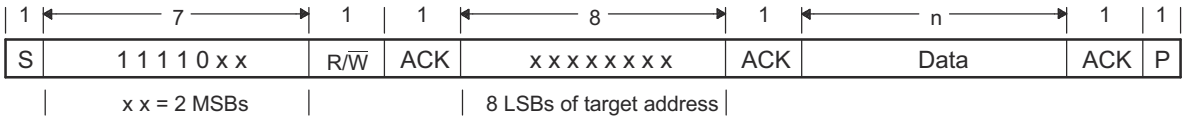


Figure 20-11. I2C Module 10-Bit Addressing Format (FDF = 0, XA = 1 in I2CMDR)

20.3.7.3 Free Data Format

The free data format can be enabled by setting I2CMDR. FDF = 1.

In this format (see Figure 20-12), the first byte after a START condition (S) is a data byte. An ACK bit is inserted after each data byte, which can be from 1 to 8 bits, depending on the BC field of I2CMDR. No address or data-direction bit is sent. Therefore, the transmitter and the receiver must both support the free data format, and the direction of the data must be constant throughout the transfer.

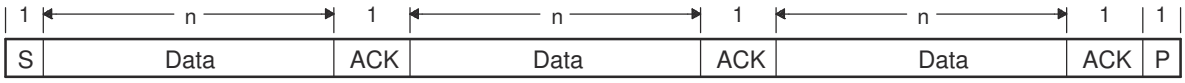


Figure 20-12. I2C Module Free Data Format (FDF = 1 in I2CMDR)

Note

The free data format is not supported in the digital loopback mode (I2CMDR.DLB = 1).

Table 20-4. How the MST and FDF Bits of I2CMDR Affect the Role of the TRX Bit of I2CMDR

MST	FDF	I2C Module State	Function of TRX
0	0	In target mode but not free data format mode	TRX is a don't care. Depending on the command from the controller, the I2C module responds as a receiver or a transmitter.
0	1	In target mode and free data format mode	The free data format mode requires that the I2C module remains the transmitter or the receiver throughout the transfer. TRX identifies the role of the I2C module: TRX = 1: The I2C module is a transmitter. TRX = 0: The I2C module is a receiver.
1	0	In controller mode but not free data format mode	TRX = 1: The I2C module is a transmitter. TRX = 0: The I2C module is a receiver.
1	1	In controller mode and free data format mode	TRX = 0: The I2C module is a receiver. TRX = 1: The I2C module is a transmitter.

20.3.7.4 Using a Repeated START Condition

I2C controller can communicate with multiple target addresses without having to give up control of the I2C bus by driving a STOP condition. This can be achieved by driving another START condition at the end of each data type. The repeated START condition can be used with the 7-bit addressing and 10-bit addressing. Figure 20-13 shows a repeated START condition in the 7-bit addressing format.

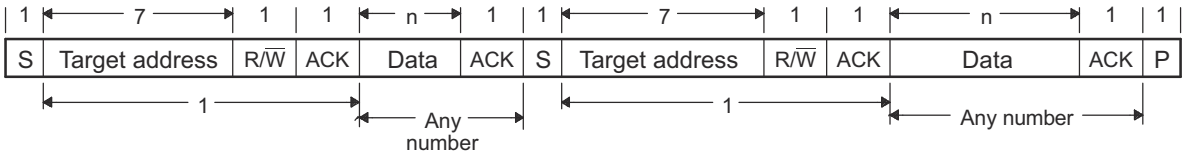


Figure 20-13. Repeated START Condition (in This Case, 7-Bit Addressing Format)

Note

In Figure 20-13, n = the number of data bits (from 1 to 8) specified by the bit count (BC) field of I2CMDR.

20.3.8 Clock Synchronization

Under normal conditions, only one controller device generates the clock signal, SCL. During the arbitration procedure, however, there are two or more controllers and the clock must be synchronized so that the data output can be compared. Figure 20-14 illustrates the clock synchronization. The wired-AND property of SCL means that a device that first generates a low period on SCL overrules the other devices. At this high-to-low transition, the clock generators of the other devices are forced to start a low period. The SCL is held low by the device with the longest low period. The other devices that finish the low periods must wait for SCL to be released, before starting the high periods. A synchronized signal on SCL is obtained, where the slowest device determines the length of the low period and the fastest device determines the length of the high period.

If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the wait state. In this way, a target slows down a fast controller and the slow device creates enough time to store a received byte or to prepare a byte to be transmitted.

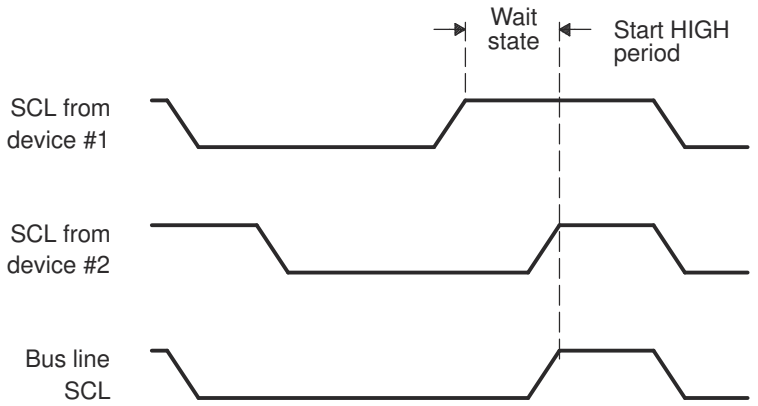


Figure 20-14. Synchronization of Two I2C Clock Generators During Arbitration

20.3.9 Clock Stretching

An I2C target device pulls the SCL line low to push the I2C controller device into the wait state preventing the I2C controller from transmitting data. This concept is called Clock Stretching.

The I2C supports three different types of clock stretching:

- **Automatic Clock Stretching:** Figure 20-15 shows the timing diagram for automatic clock stretching.
  - Target Receiver mode:
    - Non-FIFO mode: I2C clock stretches after receiving 2 bytes from the controller.
    - FIFO mode: I2C clock stretches after receiving 17 bytes from the controller.
  - Target Transmitter mode: I2C clock stretches, if I2C transmit buffer is empty.

Note

Automatic Clock Stretching is enabled by default and cannot be disabled.

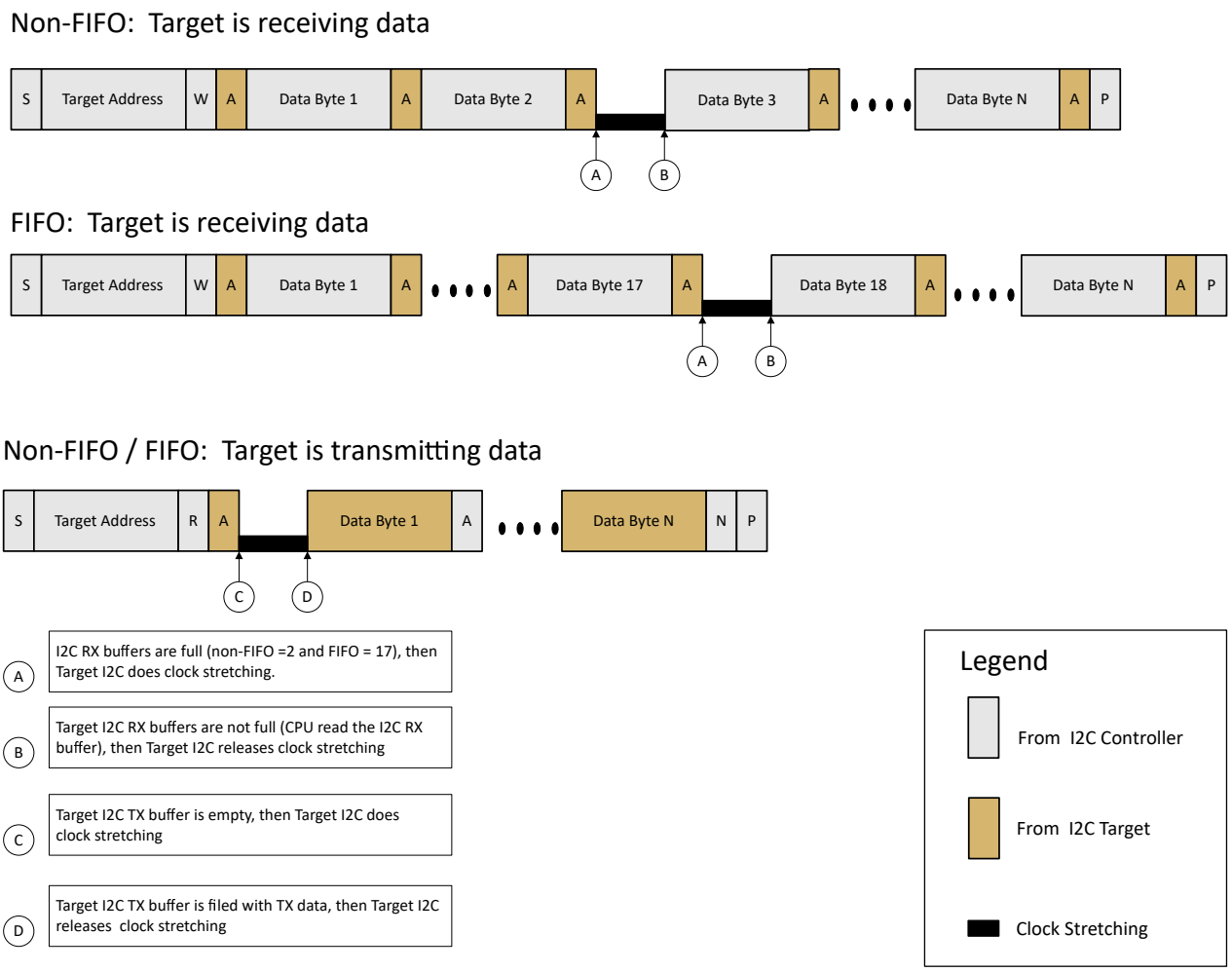


Figure 20-15. Automatic Clock Stretching

- **Extended Automatic Clock Stretching:** provides enhanced flexibility to preemptively clock stretch **after** every address and data byte, to allow the I2C target to performing housekeeping.

  Extended Automatic Clock Stretching can be enabled through the following steps:

  1. Set I2CEMDR.ALT_ECS = 1 to enable ECS.
  2. Once this feature is enabled, the hardware automatically does clock stretching by pulling the SCL line low after every ACK/NACK and generate an I2C interrupt, if using interrupts. If using the polling method, the I2CSTR.SCL_ECS flag can be used to monitor the status.
  3. Within the I2C ISR, I2CSTR.SCL_ECS is set to 1 to release the I2C from clock stretching.

  Figure 20-16 shows the timing diagram for extended automatic clock stretching.

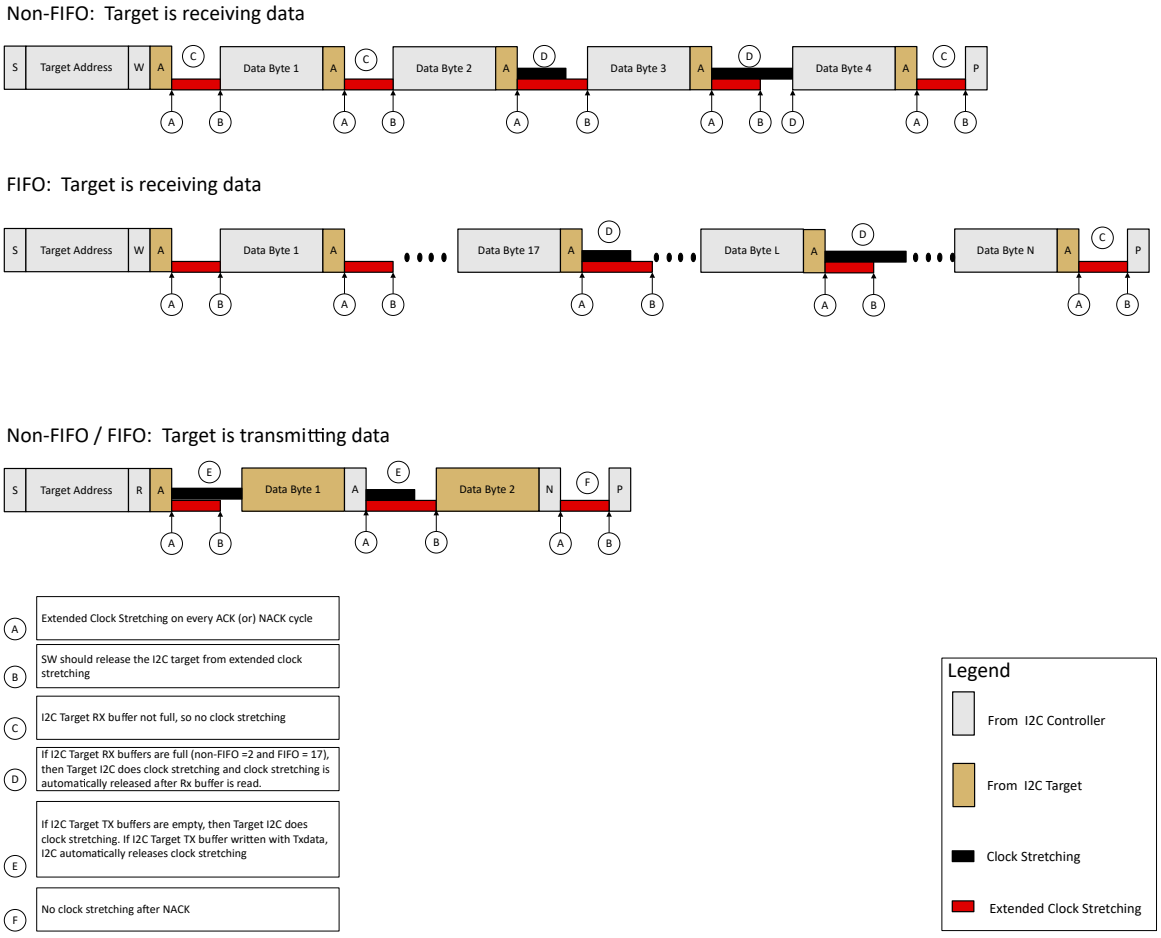


Figure 20-16. Extended Automatic Clock Stretching

- **Manual Clock Stretching:** can be enabled by setting I2CEMDR.MCS = 1 within an ISR routine. Once this feature is enabled, I2C hardware can be configured to manually do clock stretching by pulling the SCL line low after specific ACK/NACK cycles. I2C module then performs housekeeping and executes normal ISR sequence steps before disabling MCS by setting I2CEMDR.MCS = 0.

Note

Manual Clock Stretching is disabled by default and can be enabled by setting I2CEMDR.MCS = 1. Timing requirements for this type of clock stretching are not provided due to dependency on user configuration and application.

20.3.10 Arbitration

If two or more controller-transmitters attempt to start a transmission on the same bus at approximately the same time, an arbitration procedure is invoked. The arbitration procedure uses the data presented on the serial data bus (SDA) by the competing transmitters. Figure 20-17 illustrates the arbitration procedure between two devices. The first controller-transmitter that releases the SDA line high is overruled by another controller-transmitter that drives the SDA low. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. If two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

If the I2C module is the losing controller, the I2c module switches to the target-receiver mode, sets the arbitration lost (ARBL) flag, and generates the arbitration-lost interrupt request.

If during a serial transfer the arbitration procedure is still in progress when a repeated START condition or a STOP condition is transmitted to SDA, the controller-transmitters involved must send the repeated START condition or the STOP condition at the same position in the format frame. Arbitration is not allowed between:

- A repeated START condition and a data bit
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

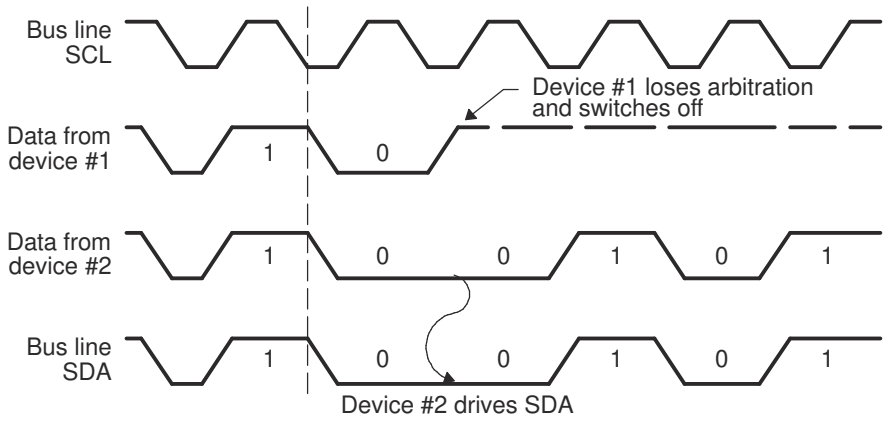


Figure 20-17. Arbitration Procedure Between Two Controller-Transmitters

20.3.11 Digital Loopback Mode

The I2C module support a self-test mode called digital loopback, which is enabled by setting the DLB bit in the I2CMDR register. In this mode, data transmitted out of the I2CDXR register is received in the I2CDRR register. The data follows an internal path, and takes n cycles to reach I2CDRR, where:

n = 8 × (SYSCLK) / (I2C module clock (Fmod))

The transmit clock and the receive clock are the same. The address seen on the external SDA pin is the address in the I2COAR register. Figure 20-18 shows the signal routing in digital loopback mode.

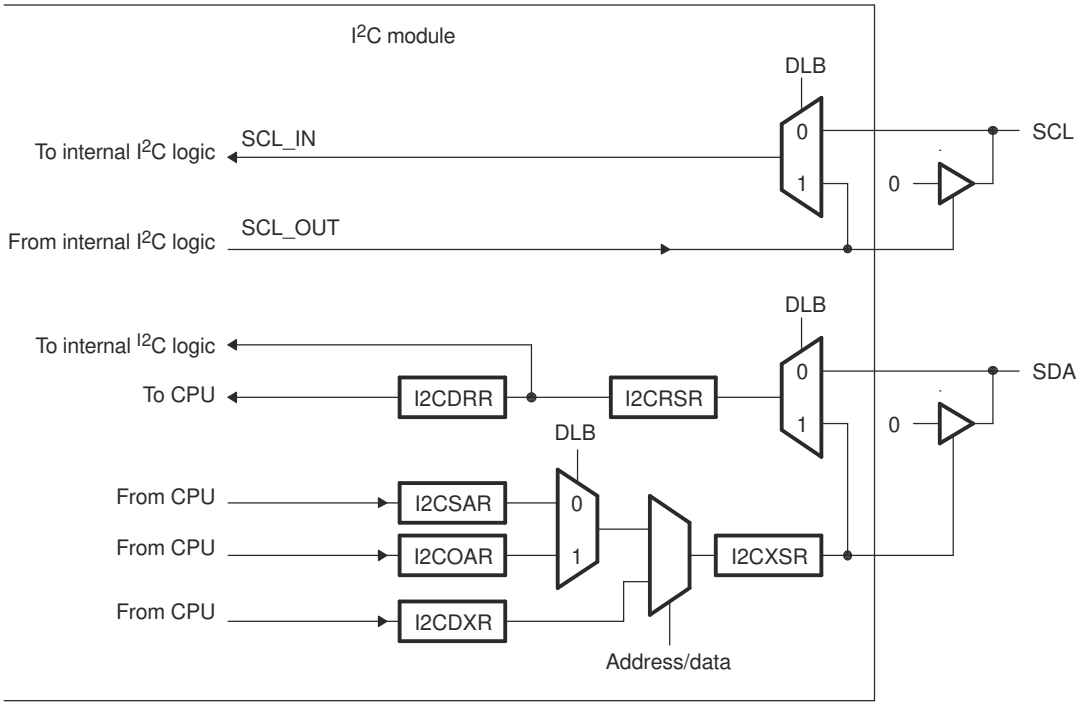


Figure 20-18. Pin Diagram Showing the Effects of the Digital Loopback Mode (DLB) Bit

Note

The free data format (I2CMDR.FDF = 1) is not supported in digital loopback mode.

20.3.12 NACK Bit Generation

When the I2C module is a receiver (controller or target), the I2C module can acknowledge or ignore bits sent by the transmitter. To ignore any new bits, the I2C module must send a no-acknowledge (NACK) bit during the acknowledge cycle on the bus. Table 20-5 summarizes the various ways you can allow the I2C module to send a NACK bit.

Note

When a NACK is sent, the following occurs:

1. The STP in I2CMDR is cleared
2. SCL is held low
3. The NACK in I2CSTR is set

Table 20-5. Ways to Generate a NACK Bit

I2C Module Condition	NACK Bit Generation Options
Target-receiver modes	Allow an overrun condition (RSFULL = 1 in I2CSTR) Reset the module (IRS = 0 in I2CMDR) Set the NACKMOD bit of I2CMDR before the rising edge of the last data bit you intend to receive
Controller-receiver mode AND Repeat mode (RM = 1 in I2CMDR)	Generate a STOP condition (STP = 1 in I2CMDR) Reset the module (IRS = 0 in I2CMDR) Set the NACKMOD bit of I2CMDR before the rising edge of the last data bit you intend to receive
Controller-receiver mode AND Nonrepeat mode (RM = 0 in I2CMDR)	If STP = 1 in I2CMDR, allow the internal data counter to count down to 0 and thus force a STOP condition If STP = 0, make STP = 1 to generate a STOP condition Reset the module (IRS = 0 in I2CMDR) Set STP = 1 to generate a STOP condition Set the NACKMOD bit of I2CMDR before the rising edge of the last data bit you intend to receive

20.4 Interrupt Requests Generated by the I2C Module

Each I2C module can generate two CPU interrupts.

1. Basic I2C interrupt: Possible basic I2C interrupt sources that can trigger this interrupt are described in Section 20.4.1.
2. I2C FIFO interrupt: Possible I2C FIFO interrupt sources that can trigger this interrupt are described in Section 20.4.2

20.4.1 Basic I2C Interrupt Requests

The I2C module generates the interrupt requests described in Table 20-6. As shown in Figure 20-19, all requests are multiplexed through an arbiter to a single I2C interrupt request to the CPU. Each interrupt request has a flag bit in the status register (I2CSTR) and an enable bit in the interrupt enable register (I2CIER). When one of the specified events occurs, the flag bit is set. If the corresponding enable bit is 0, the interrupt request is blocked. If the enable bit is 1, the request is forwarded to the CPU as an I2C interrupt.

The I2C interrupt is one of the maskable interrupts of the CPU. As with any maskable interrupt request, if the request is properly enabled in the CPU, the CPU executes the corresponding interrupt service routine (I2CINT1A_ISR). The I2CINT1A_ISR for the I2C interrupt can determine the interrupt source by reading the interrupt source register, I2CISRC. Then the I2CINT1A_ISR can branch to the appropriate subroutine.

After the CPU reads I2CISRC, the following events occur:

1. The flag for the source interrupt is cleared in I2CSTR. Exception: The ARDY, RRDY, and XRDY bits in I2CSTR are not cleared when I2CISRC is read. To clear one of these bits, write a 1 to the bit.
2. The arbiter determines which of the remaining interrupt requests has the highest priority, writes the code for that interrupt to I2CISRC, and forwards the interrupt request to the CPU.

Table 20-6. Descriptions of the Basic I2C Interrupt Requests

I2C Interrupt Request	Interrupt Source
XRDYINT	Transmit ready condition: The data transmit register (I2CDXR) is ready to accept new data because the previous data has been copied from I2CDXR to the transmit shift register (I2CXSR). As an alternative to using XRDYINT, the CPU can poll the XRDY bit of the status register, I2CSTR. XRDYINT must not be used when in FIFO mode. Use the FIFO interrupts instead.
RRDYINT	Receive ready condition: The data receive register (I2CDRR) is ready to be read because data has been copied from the receive shift register (I2CRSR) to I2CDRR. As an alternative to using RRDYINT, the CPU can poll the RRDY bit of I2CSTR. RRDYINT must not be used when in FIFO mode. Use the FIFO interrupts instead.
ARDYINT	Register-access ready condition: The I2C module registers are ready to be accessed because the previously programmed address, data, and command values have been used. The specific events that generate ARDYINT are the same events that set the ARDY bit of I2CSTR. As an alternative to using ARDYINT, the CPU can poll the ARDY bit.
NACKINT	No-acknowledgment condition: The I2C module is configured as a controller-transmitter and did not received acknowledgment from the target-receiver. As an alternative to using NACKINT, the CPU can poll the NACK bit of I2CSTR.
ARBLINT	Arbitration-lost condition: The I2C module has lost an arbitration contest with another controller-transmitter. As an alternative to using ARBLINT, the CPU can poll the ARBL bit of I2CSTR.
SCDINT	Stop condition detected: A STOP condition was detected on the I2C bus. As an alternative to using SCDINT, the CPU can poll the SCD bit of the status register, I2CSTR.
AASINT	Addressed as target condition: The I2C has been addressed as a target device by another controller on the I2C bus. As an alternative to using AASINT, the CPU can poll the AAS bit of the status register, I2CSTR.
SCLECSINT	SCL Extended Automatic Clock Stretching: SCL line is pulled low in extended automatic clock stretching mode. As an alternative to using SCLECSINT, the CPU can poll the SCL_ECS bit of the status register, I2CSTR.

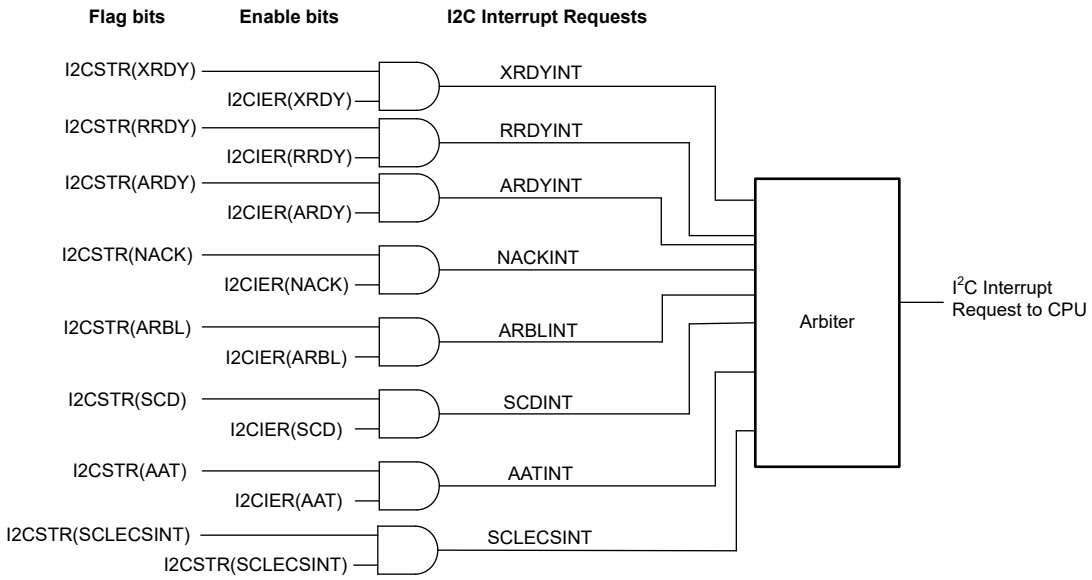


Figure 20-19. Enable Paths of the I2C Interrupt Requests

The priorities of the basic I2C interrupt requests are listed in order of highest priority to lowest priority:

1. ARBLINT
2. NACKINT
3. ARDYINT
4. RRDYINT
5. XRDYINT
6. SCDINT
7. AATINT
8. SCLECSINT

The I2C module has a backwards compatibility bit (BC) in the I2CEMDR register. The timing diagram in demonstrates the effect the backwards compatibility bit has on I2C module registers and interrupts when configured as a target-transmitter.

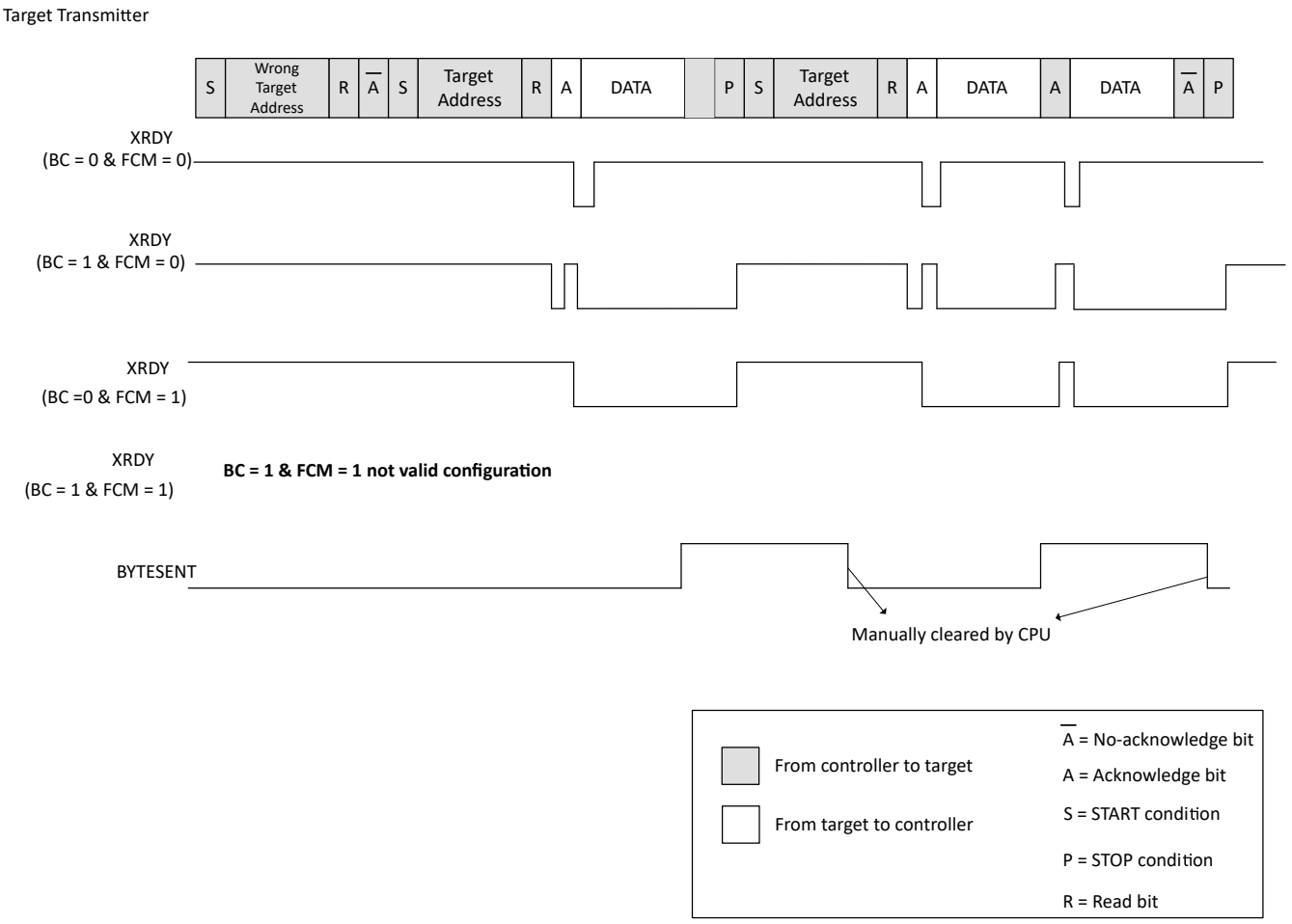


Figure 20-20. Backwards Compatibility Mode and Forward Compatibility Bit, Target Transmitter

20.4.2 I2C FIFO Interrupts

In addition to the seven basic I2C interrupts, the transmit and receive FIFOs each contain the ability to generate an interrupt (I2CINT2A). The transmit FIFO can be configured to generate an interrupt after transmitting a defined number of bytes, up to 16. The receive FIFO can be configured to generate an interrupt after receiving a defined number of bytes, up to 16. These two interrupt sources are ORed together into a single maskable CPU interrupt. Figure 20-21 shows the structure of I2C FIFO interrupt. The interrupt service routine can then read the FIFO interrupt status flags to determine from which source the interrupt came. See the I2C transmit FIFO register (I2CFFTX) and the I2C receive FIFO register (I2CFFRX) descriptions.

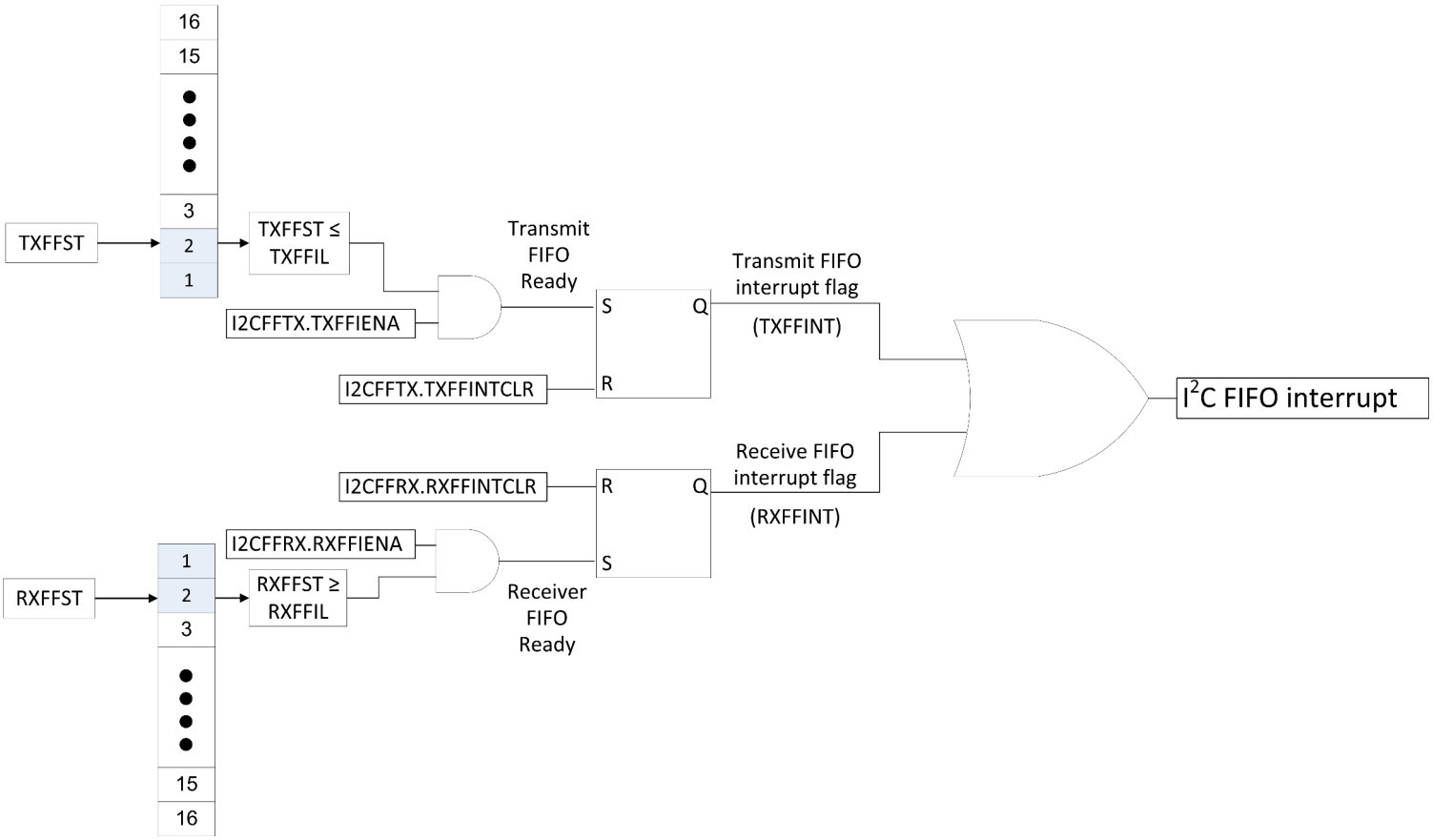


Figure 20-21. I2C FIFO Interrupt

20.5 Resetting or Disabling the I2C Module

You can reset or disable the I2C module in two ways:

- Write 0 to the I2C reset bit (IRS) in the I2C mode register (I2CMDR). All status bits (in I2CSTR) are forced to the default values, and the I2C module remains disabled until IRS is changed to 1. The SDA and SCL pins are in the high-impedance state.
- Initiate a device reset by driving the $\overline{\text{XRS}}$ pin low. The entire device is reset and is held in the reset state until you drive the pin high. When the $\overline{\text{XRS}}$ pin is released, all I2C module registers are reset to the default values. The IRS bit is forced to 0, which resets the I2C module. The I2C module stays in the reset state until you write 1 to IRS.

The IRS must be 0 while you configure or reconfigure the I2C module. Forcing IRS to 0 can be used to save power and to clear error conditions.

20.6 Software

20.6.1 I2C Registers to Driverlib Functions

Table 20-7. I2C Registers to Driverlib Functions

File	Driverlib Function
OAR	
i2c.h	I2C_setOwnAddress
IER	
i2c.c	I2C_enableInterrupt
i2c.c	I2C_disableInterrupt
STR	
i2c.c	I2C_getInterruptStatus
i2c.c	I2C_clearInterruptStatus
i2c.h	I2C_isBusBusy
i2c.h	I2C_getStatus
i2c.h	I2C_clearStatus
CLKL	
i2c.c	I2C_initController
i2c.c	I2C_initControllerModuleFrequency
CLKH	
i2c.c	I2C_initController
i2c.c	I2C_initControllerModuleFrequency
CNT	
i2c.h	I2C_setDataCount
DRR	
i2c.h	I2C_getData
TAR	
i2c.h	I2C_setTargetAddress
DXR	
i2c.h	I2C_putData
MDR	
i2c.h	I2C_enableModule
i2c.h	I2C_disableModule
i2c.h	I2C_setConfig
i2c.h	I2C_setBitCount
i2c.h	I2C_sendStartCondition
i2c.h	I2C_sendStopCondition
i2c.h	I2C_sendNACK
i2c.h	I2C_getStopConditionStatus
i2c.h	I2C_setAddressMode
i2c.h	I2C_setEmulationMode
i2c.h	I2C_enableLoopback
i2c.h	I2C_disableLoopback
ISRC	
i2c.h	I2C_getInterruptSource
EMDR	
i2c.h	I2C_setExtendedMode

Table 20-7. I2C Registers to Driverlib Functions (continued)

File	Driverlib Function
i2c.h	I2C_enableExtendedAutomaticClkStretchingMode
i2c.h	I2C_disableExtendedAutomaticClkStretchingMode
i2c.h	I2C_enableManualClkStretchingMode
i2c.h	I2C_disableManualClkStretchingMode
i2c.h	I2C_enableNACKCompatibilityMode
i2c.h	I2C_disableNACKCompatibilityMode
PSC	
i2c.c	I2C_initController
i2c.c	I2C_initControllerModuleFrequency
i2c.c	I2C_configureModuleFrequency
i2c.c	I2C_configureModuleClockFrequency
i2c.h	I2C_getPreScaler
FFTX	
i2c.c	I2C_enableInterrupt
i2c.c	I2C_disableInterrupt
i2c.c	I2C_getInterruptStatus
i2c.c	I2C_clearInterruptStatus
i2c.h	I2C_enableFIFO
i2c.h	I2C_disableFIFO
i2c.h	I2C_setFIFOInterruptLevel
i2c.h	I2C_getFIFOInterruptLevel
i2c.h	I2C_getTxFIFOStatus
FFRX	
i2c.c	I2C_enableInterrupt
i2c.c	I2C_disableInterrupt
i2c.c	I2C_getInterruptStatus
i2c.c	I2C_clearInterruptStatus
i2c.h	I2C_enableFIFO
i2c.h	I2C_disableFIFO
i2c.h	I2C_setFIFOInterruptLevel
i2c.h	I2C_getFIFOInterruptLevel
i2c.h	I2C_getRxFIFOStatus

20.6.2 I2C Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/i2c

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

20.6.2.1 C28x-I2C Library source file for FIFO interrupts

FILE: i2cLib_FIFO_controller_interrupt.c

20.6.2.2 C28x-I2C Library source file for FIFO using polling

FILE: i2cLib_FIFO_polling.c

20.6.2.3 I2C Digital Loopback with FIFO Interrupts

FILE: i2c_ex1_loopback.c

This program uses the internal loopback test mode of the I2C module. Both the TX and RX I2C FIFOs and their interrupts are used. The pinmux and I2C initialization is done through the sysconfig file.

A stream of data is sent and then compared to the received stream. The sent data looks like this:
0000 0001
0001 0002
0002 0003
....
00FE 00FF
00FF 0000
etc..
This pattern is repeated forever.

This example project has support for migration across our C2000 device families. If you are wanting to build this project from launchpad or controlCARD, please specify in the .syscfg file the board you're using. At any time you can select another device to migrate this example. *External Connections*

- None

Watch Variables

- *sData* - Data to send
- *rData* - Received data
- *rDataPoint* - Used to keep track of the last position in the receive stream for error checking

20.6.2.4 I2C EEPROM

FILE: i2c_ex2_eeprom.c

This program will write 1-14 words to EEPROM and read them back. The data written and the EEPROM address written to are contained in the message structure, i2cMsgOut. The data read back will be contained in the message structure i2cMsgIn.

External Connections on Control card

- Connect external I2C EEPROM at address 0x50
- Connect GPIO32/SDAA on controlCARD to external EEPROM SDA (serial data) pin
- Connect GPIO33/SCLA on controlCARD to external EEPROM SCL (serial clock) pin

External Connections on Launchpad

- Connect external I2C EEPROM at address 0x50
- Connect GPIO35/SDAA on Launchpad to external EEPROM SDA (serial data) pin
- Connect GPIO37/SCLA on Launchpad to external EEPROM SCL (serial clock) pin

20.6.2.5 I2C EEPROM

FILE: i2c_ex4_eeprom_polling.c

This program will shows how to perform different EEPROM write and read commands using I2C polling method EEPROM used for this example is AT24C256

External Connections

- Connect external I2C EEPROM at address 0x50 Signal | I2CA | EEPROM

SCL | DEVICE_GPIO_PIN_SCLA | SCL SDA | DEVICE_GPIO_PIN_SDAA | SDA

Make sure to connect GND pins if EEPROM and C2000 device are in different board.

20.6.2.6 I2C EEPROM

FILE: i2c_ex6_eeprom_interrupt.c

This program will shows how to perform different EEPROM write and read commands using I2C interrupts EEPROM used for this example is AT24C256

External Connections

- Connect external I2C EEPROM at address 0x50 Signal | I2CA | EEPROM

SCL | DEVICE_GPIO_PIN_SCLA | SCL SDA | DEVICE_GPIO_PIN_SDAA | SDA

Make sure to connect GND pins if EEPROM and C2000 device are in different board.

//Example 1: EEPROM Byte Write //Example 2: EEPROM Byte Read //Example 3: EEPROM word (16-bit) write //Example 4: EEPROM word (16-bit) read //Example 5: EEPROM Page write //Example 6: EEPROM word Paged read

Watch Variables

- *TX_MsgBuffer* - Message buffer which stores the data to be transmitted
- *RX_MsgBuffer* - Message buffer which stores the data to be received

20.7 I2C Registers

This section describes the I2C registers.

20.7.1 I2C Base Address Table

Table 20-8. I2C Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
I2caRegs	I2C_REGS	I2CA_BASE	0x0000_7300	-	YES

20.7.2 I2C_REGS Registers

Table 20-9 lists the memory-mapped registers for the I2C_REGS registers. All register offset addresses not listed in Table 20-9 should be considered as reserved locations and the register contents should not be modified.

Table 20-9. I2C_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	I2COAR	I2C Own address	
1h	I2CIER	I2C Interrupt Enable	
2h	I2CSTR	I2C Status	
3h	I2CCLKL	I2C Clock low-time divider	
4h	I2CCLKH	I2C Clock high-time divider	
5h	I2CCNT	I2C Data count	
6h	I2CDRR	I2C Data receive	
7h	I2CTAR	I2C TARGET address	
8h	I2CDXR	I2C Data Transmit	
9h	I2CMDR	I2C Mode	
Ah	I2CISRC	I2C Interrupt Source	
Bh	I2CEMDR	I2C Extended Mode	
Ch	I2CPSC	I2C Prescaler	
20h	I2CFFTX	I2C FIFO Transmit	
21h	I2CFFRX	I2C FIFO Receive	

Complex bit access types are encoded to fit into small table cells. Table 20-10 shows the codes that are used for access types in this section.

Table 20-10. I2C_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

20.7.2.1 I2COAR Register (Offset = 0h) [Reset = 0000h]

I2COAR is shown in Figure 20-22 and described in Table 20-11.

Return to the Summary Table.

The I2C own address register (I2COAR) is a 16-bit register. The I2C module uses this register to specify its own TARGET address, which distinguishes it from other TARGETs connected to the I2C-bus. If the 7-bit addressing mode is selected (XA = 0 in I2CMDR), only bits 6-0 are used
write 0s to bits 9-7.

Figure 20-22. I2COAR Register

15	14	13	12	11	10	9	8
RESERVED						OAR	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OAR							
R/W-0h							

Table 20-11. I2COAR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	OAR	R/W	0h	In 7-bit addressing mode (XA = 0 in I2CMDR): 00h-7Fh Bits 6-0 provide the 7-bit TARGET address of the I2C module. Write 0s to bits 9-7. In 10-bit addressing mode (XA = 1 in I2CMDR): 000h-3FFh Bits 9-0 provide the 10-bit TARGET address of the I2C module. Reset type: SYSRSn

20.7.2.2 I2CIER Register (Offset = 1h) [Reset = 0000h]

I2CIER is shown in Figure 20-23 and described in Table 20-12.

Return to the Summary Table.

I2CIER is used by the CPU to individually enable or disable I2C interrupt requests.

Figure 20-23. I2CIER Register

15	14	13	12	11	10	9	8
SCL_ECS	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED	AAT	SCD	XRDY	RRDY	ARDY	NACK	ARBL
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-12. I2CIER Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SCL_ECS	R/W	0h	SCL Extended Automatic Clock Stretch interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
14-7	RESERVED	R	0h	Reserved
6	AAT	R/W	0h	Addressed as TARGET interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
5	SCD	R/W	0h	Stop condition detected interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
4	XRDY	R/W	0h	Transmit-data-ready interrupt enable bit. This bit should not be set when using FIFO mode. Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
3	RRDY	R/W	0h	Receive-data-ready interrupt enable bit. This bit should not be set when using FIFO mode. Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
2	ARDY	R/W	0h	Register-access-ready interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
1	NACK	R/W	0h	No-acknowledgment interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled
0	ARBL	R/W	0h	Arbitration-lost interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt request disabled 1h (R/W) = Interrupt request enabled

20.7.2.3 I2CSTR Register (Offset = 2h) [Reset = 0410h]

I2CSTR is shown in Figure 20-24 and described in Table 20-13.

Return to the Summary Table.

The I2C status register (I2CSTR) is a 16-bit register used to determine which interrupt has occurred and to read status information.

Figure 20-24. I2CSTR Register

15	14	13	12	11	10	9	8
SCL_ECS	TDIR	NACKSNT	BB	RSFULL	XSMT	AAT	AD0
R/W1C-0h	R/W1C-0h	R/W1C-0h	R-0h	R-0h	R-1h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	BYTESENT	SCD	XRDY	RRDY	ARDY	NACK	ARBL
R-0h	R/W1C-0h	R/W1C-0h	R-1h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 20-13. I2CSTR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SCL_ECS	R/W1C	0h	SCL Status bit in extended automatic clock stretching mode 0: SCL line is pulled high. 1: SCL line is pulled low after ACK (or) NACK phase. Writing '1' to this bit releases SCL. SCL line is pulled high. Note: This bit is applicable only for extended clock stretching mode Reset type: SYSRSn 0h (R/W) = SCL line is not pulled low in extended automatic clock stretching mode. SCL_ECS bit is cleared by one o fthe following events: - It is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = SCL line is pulled low in extended automatic clock stretching mode.
14	TDIR	R/W1C	0h	TARGET direction bit Reset type: SYSRSn 0h (R/W) = I2C is not addressed as a TARGET transmitter. TDIR is cleared by one of the following events: - It is manually cleared. To clear this bit, write a 1 to it. - Digital loopback mode is enabled. - A START or STOP condition occurs on the I2C bus. 1h (R/W) = I2C is addressed as a TARGET transmitter.
13	NACKSNT	R/W1C	0h	NACK sent bit. This bit is used when the I2C module is in the receiver mode. One instance in which NACKSNT is affected is when the NACK mode is used (see the description for NACKMOD in Reset type: SYSRSn 0h (R/W) = NACK not sent. NACKSNT bit is cleared by any one of the following events: - It is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset (either when 0 is written to the IRS bit of I2CMDR or when the whole device is reset). 1h (R/W) = NACK sent: A no-acknowledge bit was sent during the acknowledge cycle on the I2C-bus.

Table 20-13. I2CSTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	BB	R	0h	Bus busy bit. BB indicates whether the I2C-bus is busy or is free for another data transfer. See the paragraph following the table for more information Reset type: SYSRSn 0h (R/W) = Bus free. BB is cleared by any one of the following events: - The I2C module receives or transmits a STOP bit (bus free). - The I2C module is reset. 1h (R/W) = Bus busy: The I2C module has received or transmitted a START bit on the bus.
11	RSFULL	R	0h	Receive shift register full bit. RSFULL indicates an overrun condition during reception. Overrun occurs when new data is received into the shift register (I2CRSR) and the old data has not been read from the receive register (I2CDRR). As new bits arrive from the SDA pin, they overwrite the bits in I2CRSR. The new data will not be copied to ICDRR until the previous data is read. Reset type: SYSRSn 0h (R/W) = No overrun detected. RSFULL is cleared by any one of the following events: - I2CDRR is read by the CPU. Emulator reads of the I2CDRR do not affect this bit. - The I2C module is reset. 1h (R/W) = Overrun detected
10	XSMT	R	1h	Transmit shift register empty bit. XSMT = 0 indicates that the transmitter has experienced underflow. Underflow occurs when the transmit shift register (I2CXSR) is empty but the data transmit register (I2CDXR) has not been loaded since the last I2CDXR-to-I2CXSR transfer. The next I2CDXR-to-I2CXSR transfer will not occur until new data is in I2CDXR. If new data is not transferred in time, the previous data may be re-transmitted on the SDA pin. Reset type: SYSRSn 0h (R/W) = Underflow detected (empty) 1h (R/W) = No underflow detected (not empty). XSMT is set by one of the following events: - Data is written to I2CDXR. - The I2C module is reset
9	AAT	R	0h	Addressed-as-TARGET bit Reset type: SYSRSn 0h (R/W) = In the 7-bit addressing mode, the AAT bit is cleared when receiving a NACK, a STOP condition, or a repeated START condition. In the 10-bit addressing mode, the AAT bit is cleared when receiving a NACK, a STOP condition, or by a TARGET address different from the I2C peripheral's own TARGET address. 1h (R/W) = The I2C module has recognized its own TARGET address or an address of all zeros (general call).
8	AD0	R	0h	Address 0 bits Reset type: SYSRSn 0h (R/W) = AD0 has been cleared by a START or STOP condition. 1h (R/W) = An address of all zeros (general call) is detected.
7	RESERVED	R	0h	Reserved

Table 20-13. I2CSTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	BYTESENT	R/W1C	0h	Byte Transmit over indication. BYTESENT is set when the CONTROLLER/TARGET has successfully sent the byte on SCL/SDA lines. This is diagnostic register which needs to be explicitly cleared by Software. In case not cleared the stale status would keep reflecting as no automated clear incorporated to avoid corner conditions. Reset type: SYSRSn 0h (R/W) = The I2C module has not finished transmitting the next data byte. BYTESENT is cleared by any one of the following events: - It is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = The I2C module has completed the transmission of a byte.
5	SCD	R/W1C	0h	Stop condition detected bit. SCD is set when the I2C sends or receives a STOP condition. The I2C module delays clearing of the I2CMDR[STP] bit until the SCD bit is set. Reset type: SYSRSn 0h (R/W) = STOP condition not detected since SCD was last cleared. SCD is cleared by any one of the following events: - I2CISRC is read by the CPU when it contains the value 110b (stop condition detected). Emulator reads of the I2CISRC do not affect this bit. - SCD is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = A STOP condition has been detected on the I2C bus.
4	XRDY	R	1h	Transmit-data-ready interrupt flag bit. When not in FIFO mode, XRDY indicates that the data transmit register (I2CDXR) is ready to accept new data. FCM=0 : When the previous data has been copied from I2CDXR to the transmit shift register (I2CXSR). The CPU can poll XRDY or use the XRDY interrupt request When in FIFO mode, use TXFFINT instead. FCM=1: XRDY is asserted only when next data is required it gets deasserted with write to I2CDXR. Both Polling and interrupt based data transfers are allowed in the FCM mode. Reset type: SYSRSn 0h (R/W) = I2CDXR not ready. XRDY is cleared when data is written to I2CDXR. 1h (R/W) = I2CDXR ready: Data has been copied from I2CDXR to I2CXSR. XRDY is also forced to 1 when the I2C module is reset.
3	RRDY	R/W1C	0h	Receive-data-ready interrupt flag bit. When not in FIFO mode, RRDY indicates that the data receive register (I2CDRR) is ready to be read because data has been copied from the receive shift register (I2CRSR) to I2CDRR. The CPU can poll RRDY or use the RRDY interrupt request When in FIFO mode, use RXFFINT instead. Reset type: SYSRSn 0h (R/W) = I2CDRR not ready. RRDY is cleared by any one of the following events: - I2CDRR is read by the CPU. Emulator reads of the I2CDRR do not affect this bit. - RRDY is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = I2CDRR ready: Data has been copied from I2CRSR to I2CDRR.

Table 20-13. I2CSTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ARDY	R/W1C	0h	Register-access-ready interrupt flag bit (only Applicable when the I2C module is in the CONTROLLER mode). ARDY indicates that the I2C module registers are ready to be accessed because the previously programmed address, data, and command values have been used. The CPU can poll ARDY or use the ARDY interrupt request Reset type: SYSRSn 0h (R/W) = The registers are not ready to be accessed. ARDY is cleared by any one of the following events: - The I2C module starts using the current register contents. - ARDY is manually cleared. To clear this bit, write a 1 to it. - The I2C module is reset. 1h (R/W) = The registers are ready to be accessed. In the nonrepeat mode (RM = 0 in I2CMDR): If STP = 0 in I2CMDR, the ARDY bit is set when the internal data counter counts down to 0. If STP = 1, ARDY is not affected (instead, the I2C module generates a STOP condition when the counter reaches 0). In the repeat mode (RM = 1): ARDY is set at the end of each byte transmitted from I2CDXR.
1	NACK	R/W1C	0h	No-acknowledgment interrupt flag bit. NACK applies when the I2C module is a CONTROLLER transmitter (or) TARGET transmitter. NACK indicates whether the I2C module has detected an acknowledge bit (ACK) or a noacknowledge bit (NACK). The CPU can poll NACK or use the NACK interrupt request. Reset type: SYSRSn 0h (R/W) = ACK received/NACK not received. This bit is cleared by any one of the following events: - An acknowledge bit (ACK) has been sent by the TARGET receiver. - NACK is manually cleared. To clear this bit, write a 1 to it. - The CPU reads the interrupt source register (I2CISRC) and the register contains the code for a NACK interrupt. Emulator reads of the I2CISRC do not affect this bit. - The I2C module is reset. 1h (R/W) = NACK bit received. The hardware detects that a no-acknowledge (NACK) bit has been received. Note: While the I2C module performs a general call transfer, NACK is 1, even if one or more TARGETs send acknowledgment.
0	ARBL	R/W1C	0h	Arbitration-lost interrupt flag bit (only applicable when the I2C module is a CONTROLLER-transmitter). ARBL primarily indicates when the I2C module has lost an arbitration contest with another CONTROLLERtransmitter. The CPU can poll ARBL or use the ARBL interrupt request. Reset type: SYSRSn 0h (R/W) = Arbitration not lost. AL is cleared by any one of the following events: - AL is manually cleared. To clear this bit, write a 1 to it. - The CPU reads the interrupt source register (I2CISRC) and the register contains the code for an AL interrupt. Emulator reads of the I2CISRC do not affect this bit. - The I2C module is reset. 1h (R/W) = Arbitration lost. AL is set by any one of the following events: - The I2C module senses that it has lost an arbitration with two or more competing transmitters that started a transmission almost simultaneously. - The I2C module attempts to start a transfer while the BB (bus busy) bit is set to 1. When AL becomes 1, the CNT and STP bits of I2CMDR are cleared, and the I2C module becomes a TARGET-receiver.

20.7.2.4 I2CCLKL Register (Offset = 3h) [Reset = 0000h]

I2CCLKL is shown in Figure 20-25 and described in Table 20-14.

Return to the Summary Table.

I2C Clock low-time divider

Figure 20-25. I2CCLKL Register

15	14	13	12	11	10	9	8
I2CCLKL							
R/W-0h							
7	6	5	4	3	2	1	0
I2CCLKL							
R/W-0h							

Table 20-14. I2CCLKL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	I2CCLKL	R/W	0h	Clock low-time divide-down value. To produce the low time duration of the CONTROLLER clock, the period of the module clock is multiplied by (ICCL + d). d is an adjustment factor based on the prescaler. See the Clock Divider Registers section of the Introduction for details. Note: These bits must be set to a non-zero value for proper I2C clock generation. Reset type: SYSRSn

20.7.2.5 I2CCLKH Register (Offset = 4h) [Reset = 0000h]

I2CCLKH is shown in Figure 20-26 and described in Table 20-15.

Return to the Summary Table.

I2C Clock high-time divider

Figure 20-26. I2CCLKH Register

15	14	13	12	11	10	9	8
I2CCLKH							
R/W-0h							
7	6	5	4	3	2	1	0
I2CCLKH							
R/W-0h							

Table 20-15. I2CCLKH Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	I2CCLKH	R/W	0h	Clock high-time divide-down value. To produce the high time duration of the CONTROLLER clock, the period of the module clock is multiplied by (ICCL + d). d is an adjustment factor based on the prescaler. See the Clock Divider Registers section of the Introduction for details. Note: These bits must be set to a non-zero value for proper I2C clock generation. Reset type: SYSRSn

20.7.2.6 I2CCNT Register (Offset = 5h) [Reset = 0000h]

I2CCNT is shown in Figure 20-27 and described in Table 20-16.

Return to the Summary Table.

I2CCNT is a 16-bit register used to indicate how many data bytes to transfer when the I2C module is configured as a transmitter, or to receive when configured as a CONTROLLER receiver. In the repeat mode (RM = 1), I2CCNT is not used.
The value written to I2CCNT is copied to an internal data counter. The internal data counter is decremented by 1 for each byte transferred (I2CCNT remains unchanged). If a STOP condition is requested in the CONTROLLER mode (STP = 1 in I2CMDR), the I2C module terminates the transfer with a STOP condition when the countdown is complete (that is, when the last byte has been transferred).

Figure 20-27. I2CCNT Register

15	14	13	12	11	10	9	8
I2CCNT							
R/W-0h							
7	6	5	4	3	2	1	0
I2CCNT							
R/W-0h							

Table 20-16. I2CCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	I2CCNT	R/W	0h	Data count value. I2CCNT indicates the number of data bytes to transfer or receive. If a STOP condition is specified (STP=1) then I2CCNT will decrease after each byte is sent until it reaches zero, which in turn will generate a STOP condition. The value in I2CCNT is a don't care when the RM bit in I2CMDR is set to 1. Reset type: SYSRSn 0h (R/W) = data count value is 65536 1h (R/W) = data count value is 1 2h (R/W) = data count value is 2 FFFFh (R/W) = data count value is 65535

20.7.2.7 I2CDRR Register (Offset = 6h) [Reset = 0000h]

I2CDRR is shown in Figure 20-28 and described in Table 20-17.

Return to the Summary Table.

I2CDRR is a 16-bit register used by the CPU to read received data. The I2C module can receive a data byte with 1 to 8 bits. The number of bits is selected with the bit count (BC) bits in I2CMDR. One bit at a time is shifted in from the SDA pin to the receive shift register (I2CRSR). When a complete data byte has been received, the I2C module copies the data byte from I2CRSR to I2CDRR. The CPU cannot access I2CRSR directly.
If a data byte with fewer than 8 bits is in I2CDRR, the data value is right-justified, and the other bits of I2CDRR(7-0) are undefined. For example, if BC = 011 (3-bit data size), the receive data is in I2CDRR(2-0), and the content of I2CDRR(7-3) is undefined.
When in the receive FIFO mode, the I2CDRR register acts as the receive FIFO buffer.

Figure 20-28. I2CDRR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
DATA							
R-0h							

Table 20-17. I2CDRR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	DATA	R	0h	Receive data Reset type: SYSRSn

20.7.2.8 I2CTAR Register (Offset = 7h) [Reset = 03FFh]

I2CTAR is shown in Figure 20-29 and described in Table 20-18.

Return to the Summary Table.

The I2C TARGET address register (I2CSAR) is a 16-bit register for storing the next TARGET address that will be transmitted by the I2C module when it is a CONTROLLER. The SAR field of I2CSAR contains a 7-bit or 10-bit TARGET address. When the I2C module is not using the free data format (FDF = 0 in I2CMDR), it uses this address to initiate data transfers with a TARGET, or TARGETs. When the address is nonzero, the address is for a particular TARGET. When the address is 0, the address is a general call to all TARGETs. If the 7-bit addressing mode is selected (XA = 0 in I2CMDR), only bits 6-0 of I2CSAR are used
write 0s to bits 9-7.

Figure 20-29. I2CTAR Register

15	14	13	12	11	10	9	8
RESERVED						TAR	
R-0h						R/W-3FFh	
7	6	5	4	3	2	1	0
TAR							
R/W-3FFh							

Table 20-18. I2CTAR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	TAR	R/W	3FFh	In 7-bit addressing mode (XA = 0 in I2CMDR): 00h-7Fh Bits 6-0 provide the 7-bit TARGET address that the I2C module transmits when it is in the CONTROLLER-transmitter mode. Write 0s to bits 9-7. In 10-bit addressing mode (XA = 1 in I2CMDR): 000h-3FFh Bits 9-0 provide the 10-bit TARGET address that the I2C module transmits when it is in the CONTROLLER transmitter mode. Reset type: SYSRSn

20.7.2.9 I2CDXR Register (Offset = 8h) [Reset = 0000h]

I2CDXR is shown in Figure 20-30 and described in Table 20-19.

Return to the Summary Table.

The CPU writes transmit data to I2CDXR. This 16-bit register accepts a data byte with 1 to 8 bits. Before writing to I2CDXR, specify how many bits are in a data byte by loading the appropriate value into the bit count (BC) bits of I2CMDR. When writing a data byte with fewer than 8 bits, make sure the value is right-aligned in I2CDXR. After a data byte is written to I2CDXR, the I2C module copies the data byte to the transmit shift register (I2CXSR). The CPU cannot access I2CXSR directly. From I2CXSR, the I2C module shifts the data byte out on the SDA pin, one bit at a time.
When in the transmit FIFO mode, the I2CDXR register acts as the transmit FIFO buffer.

Figure 20-30. I2CDXR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
DATA							
R/W-0h							

Table 20-19. I2CDXR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	DATA	R/W	0h	Transmit data Reset type: SYSRSn

20.7.2.10 I2CMDR Register (Offset = 9h) [Reset = 0000h]

I2CMDR is shown in Figure 20-31 and described in Table 20-20.

Return to the Summary Table.

The I2C mode register (I2CMDR) is a 16-bit register that contains the control bits of the I2C module.

Figure 20-31. I2CMDR Register

15	14	13	12	11	10	9	8
NACKMOD	FREE	STT	RESERVED	STP	CNT	TRX	XA
R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RM	DLB	IRS	STB	FDF	BC		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		

Table 20-20. I2CMDR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	NACKMOD	R/W	0h	NACK mode bit. This bit is only applicable when the I2C module is acting as a receiver. Reset type: SYSRSn 0h (R/W) = In the TARGET-receiver mode: The I2C module sends an acknowledge (ACK) bit to the transmitter during each acknowledge cycle on the bus. The I2C module only sends a no-acknowledge (NACK) bit if you set the NACKMOD bit. In the CONTROLLER-receiver mode: The I2C module sends an ACK bit during each acknowledge cycle until the internal data counter counts down to 0. At that point, the I2C module sends a NACK bit to the transmitter. To have a NACK bit sent earlier, you must set the NACKMOD bit 1h (R/W) = In either TARGET-receiver or CONTROLLER-receiver mode: The I2C module sends a NACK bit to the transmitter during the next acknowledge cycle on the bus. Once the NACK bit has been sent, NACKMOD is cleared. Important: To send a NACK bit in the next acknowledge cycle, you must set NACKMOD before the rising edge of the last data bit.
14	FREE	R/W	0h	This bit controls the action taken by the I2C module when a debugger breakpoint is encountered. Reset type: SYSRSn 0h (R/W) = When I2C module is CONTROLLER: If SCL is low when the breakpoint occurs, the I2C module stops immediately and keeps driving SCL low, whether the I2C module is the transmitter or the receiver. If SCL is high, the I2C module waits until SCL becomes low and then stops. When I2C module is TARGET: A breakpoint forces the I2C module to stop when the current transmission/reception is complete. 1h (R/W) = The I2C module runs free that is, it continues to operate when a breakpoint occurs.
13	STT	R/W	0h	START condition bit (only applicable when the I2C module is a CONTROLLER). The RM, STT, and STP bits determine when the I2C module starts and stops data transmissions (see Table 9-6). Note that the STT and STP bits can be used to terminate the repeat mode, and that this bit is not writable when IRS = 0. Reset type: SYSRSn 0h (R/W) = In the CONTROLLER mode, STT is automatically cleared after the START condition has been generated. 1h (R/W) = In the CONTROLLER mode, setting STT to 1 causes the I2C module to generate a START condition on the I2C-bus
12	RESERVED	R	0h	Reserved

Table 20-20. I2CMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	STP	R/W	0h	STOP condition bit (only applicable when the I2C module is a CONTROLLER). In the CONTROLLER mode, the RM,STT, and STP bits determine when the I2C module starts and stops data transmissions. Note that the STT and STP bits can be used to terminate the repeat mode, and that this bit is not writable when IRS=0. When in non-repeat mode, at least one byte must be transferred before a stop condition can be generated. The I2C module delays clearing of this bit until after the I2CSTR[SCD] bit is set. To avoid disrupting the I2C state machine, the user must wait until this bit is clear before initiating a new message. Reset type: SYSRSn 0h (R/W) = STP is automatically cleared after the STOP condition has been generated 1h (R/W) = STP has been set by the device to generate a STOP condition when the internal data counter of the I2C module counts down to 0.
10	CNT	R/W	0h	CONTROLLER mode bit. CNT determines whether the I2C module is in the TARGET mode or the CONTROLLER mode. CNT is automatically changed from 1 to 0 when the I2C CONTROLLER generates a STOP condition Reset type: SYSRSn 0h (R/W) = TARGET mode. The I2C module is a TARGET and receives the serial clock from the CONTROLLER. 1h (R/W) = CONTROLLER mode. The I2C module is a CONTROLLER and generates the serial clock on the SCL pin.
9	TRX	R/W	0h	Transmitter mode bit. When relevant, TRX selects whether the I2C module is in the transmitter mode or the receiver mode. Reset type: SYSRSn 0h (R/W) = Receiver mode. The I2C module is a receiver and receives data on the SDA pin. 1h (R/W) = Transmitter mode. The I2C module is a transmitter and transmits data on the SDA pin.
8	XA	R/W	0h	Expanded address enable bit. Reset type: SYSRSn 0h (R/W) = 7-bit addressing mode (normal address mode). The I2C module transmits 7-bit TARGET addresses (from bits 6-0 of I2CTAR), and its own TARGET address has 7 bits (bits 6-0 of I2COAR). 1h (R/W) = 10-bit addressing mode (expanded address mode). The I2C module transmits 10-bit TARGET addresses (from bits 9-0 of I2CTAR), and its own TARGET address has 10 bits (bits 9-0 of I2COAR).
7	RM	R/W	0h	Repeat mode bit (only applicable when the I2C module is a CONTROLLER-transmitter or CONTROLLER-receiver). The RM, STT, and STP bits determine when the I2C module starts and stops data transmissions Reset type: SYSRSn 0h (R/W) = Nonrepeat mode. The value in the data count register (I2CCNT) determines how many bytes are received/transmitted by the I2C module. 1h (R/W) = Repeat mode. A data byte is transmitted each time the I2CDXR register is written to (or until the transmit FIFO is empty when in FIFO mode) until the STP bit is manually set. The value of I2CCNT is ignored. The ARDY bit/interrupt can be used to determine when the I2CDXR (or FIFO) is ready for more data, or when the data has all been sent and the CPU is allowed to write to the STP bit.

Table 20-20. I2CMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	DLB	R/W	0h	Digital loopback mode bit. Reset type: SYSRSn 0h (R/W) = Digital loopback mode is disabled. 1h (R/W) = Digital loopback mode is enabled. For proper operation in this mode, the CNT bit must be 1. In the digital loopback mode, data transmitted out of I2CDXR is received in I2CDRR after n device cycles by an internal path, where: n = ((I2C input clock frequency/module clock frequency) x 8) The transmit clock is also the receive clock. The address transmitted on the SDA pin is the address in I2COAR. Note: The free data format (FDF = 1) is not supported in the digital loopback mode.
5	IRS	R/W	0h	I2C module reset bit. Reset type: SYSRSn 0h (R/W) = The I2C module is in reset/disabled. When this bit is cleared to 0, all status bits (in I2CSTR) are set to their default values. 1h (R/W) = The I2C module is enabled. This has the effect of releasing the I2C bus if the I2C peripheral is holding it.
4	STB	R/W	0h	START byte mode bit. This bit is only applicable when the I2C module is a CONTROLLER. As described in version 2.1 of the Philips Semiconductors I2C-bus specification, the START byte can be used to help a TARGET that needs extra time to detect a START condition. When the I2C module is a TARGET, it ignores a START byte from a CONTROLLER, regardless of the value of the STB bit. Reset type: SYSRSn 0h (R/W) = The I2C module is not in the START byte mode. 1h (R/W) = The I2C module is in the START byte mode. When you set the START condition bit (STT), the I2C module begins the transfer with more than just a START condition. Specifically, it generates: 1. A START condition 2. A START byte (0000 0001b) 3. A dummy acknowledge clock pulse 4. A repeated START condition Then, as normal, the I2C module sends the TARGET address that is in I2CTAR.
3	FDF	R/W	0h	Free data format mode bit. Reset type: SYSRSn 0h (R/W) = Free data format mode is disabled. Transfers use the 7-/10-bit addressing format selected by the XA bit. 1h (R/W) = Free data format mode is enabled. Transfers have the free data (no address) format described in Section 9.2.5. The free data format is not supported in the digital loopback mode (DLB=1).

Table 20-20. I2CMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	BC	R/W	0h	Bit count bits. BC defines the number of bits (1 to 8) in the next data byte that is to be received or transmitted by the I2C module. The number of bits selected with BC must match the data size of the other device. Notice that when BC = 000b, a data byte has 8 bits. BC does not affect address bytes, which always have 8 bits. Note: If the bit count is less than 8, receive data is right-justified in I2CDRR(7-0), and the other bits of I2CDRR(7-0) are undefined. Also, transmit data written to I2CDXR must be right-justified Reset type: SYSRSn 0h (R/W) = 8 bits per data byte 1h (R/W) = 1 bit per data byte 2h (R/W) = 2 bits per data byte 3h (R/W) = 3 bits per data byte 4h (R/W) = 4 bits per data byte 5h (R/W) = 5 bits per data byte 6h (R/W) = 6 bits per data byte 7h (R/W) = 7 bits per data byte

20.7.2.11 I2CISRC Register (Offset = Ah) [Reset = 0000h]

I2CISRC is shown in Figure 20-32 and described in Table 20-21.

Return to the Summary Table.

The I2C interrupt source register (I2CISRC) is a 16-bit register used by the CPU to determine which event generated the I2C interrupt.

Figure 20-32. I2CISRC Register

15	14	13	12	11	10	9	8
RESERVED				WRITE_ZEROS			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED				INTCODE			
R-0h				R-0h			

Table 20-21. I2CISRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-8	WRITE_ZEROS	R/W	0h	TI internal testing bits These reserved bit locations should always be written as zeros. Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3-0	INTCODE	R	0h	Interrupt code bits. The binary code in INTCODE indicates the event that generated an I2C interrupt. A CPU read will clear this field. If another lower priority interrupt is pending and enabled, the value corresponding to that interrupt will then be loaded. Otherwise, the value will stay cleared. The interrupt events below are listed in descending order of priority. That is INTCODE 1 (Arbitration lost) has the highest priority and INTCODE 7 (Addressed as TARGET) has the lowest priority. In the case of an arbitration lost, a no-acknowledgment condition detected, or a stop condition detected, a CPU read will also clear the associated interrupt flag bit in the I2CSTR register. Emulator reads will not affect the state of this field or of the status bits in the I2CSTR register. Reset type: SYSRSn 0h (R/W) = None 1h (R/W) = Arbitration lost 2h (R/W) = No-acknowledgment condition detected 3h (R/W) = Registers ready to be accessed 4h (R/W) = Receive data ready 5h (R/W) = Transmit data ready 6h (R/W) = Stop condition detected 7h (R/W) = Addressed as TARGET 8h (R/W) = Extended Automatic Clock Stretching

20.7.2.12 I2CEMDR Register (Offset = Bh) [Reset = 0001h]

I2CEMDR is shown in Figure 20-33 and described in Table 20-22.

Return to the Summary Table.

The I2CEMDR is a 16-bit register that enables additional I2C features such as forward / backward compatibility and types of clock stretching.

Figure 20-33. I2CEMDR Register

15	14	13	12	11	10	9	8
RESERVED	RESERVED						NACK_CM
R/W-0h	R-0h						R/W-0h
7	6	5	4	3	2	1	0
SCLKEY				MCS	ECS	FCM	BC
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-1h

Table 20-22. I2CEMDR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R/W	0h	Reserved
14-9	RESERVED	R	0h	Reserved
8	NACK_CM	R/W	0h	NACK Compatibility mode 0: I2CSTR.NACK bit and NACK interrupt gets triggered when NACK is received in Controller Transmitter mode. 1: I2CSTR.NACK bit and NACK interrupt gets triggered when NACK is received in both Controller Transmitter mode and Target Transmitter mode Reset type: SYSRSn
7-4	SCLKEY	R/W	0h	Clock Stretching Key. 0xA: Writes to all bits related to Clock Stretching modes in this register are enabled. Other Values: Writes to all bits related to Clock Stretching modes in this register are ignored. Reset type: SYSRSn
3	MCS	R/W	0h	Manual Clock Stretching mode 0: Manual Clock Stretching is disabled and SCL line is pulled high. 1: Manual Clock Stretching is enabled and SCL line is pulled low, allowing software to control clock stretching before or after every ACK / NACK cycle. Note: When SCLKEY is not set to 0xA, Manual Clock Stretching mode will remain disabled. Note: MCS should be disabled when other clock stretching modes are enabled to prevent a bus hang. Reset type: SYSRSn
2	ECS	R/W	0h	Extended Automatic Clock Stretching mode 0: Extended Automatic Clock Stretching feature is disabled 1: Extended Automatic Clock Stretching feature is enabled. When enabled, I2C target automatically clock stretches after every ACK / NACK cycle. Note: When SCLKEY is not set to 0xA, Extended Automatic Clock Stretching mode will remain disabled. Reset type: SYSRSn
1	FCM	R/W	0h	Forward Compatibility mode. This bit when programmed brings the functionality of Tx request only when Tx data required regardless of data status in Tx buffer for non-FIFO mode. This register affects the XRDY behavior hence needs to be set after releasing the IRS (I2CMDR[5]). Reset type: SYSRSn

Table 20-22. I2CEMDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	BC	R/W	1h	Backwards compatibility mode. This bit affects the timing of the transmit status bits (XRDY and XSMT) in the I2CSTR register when in TARGET transmitter mode. Reset type: SYSRSn 0h (R/W) = See the 'Backwards Compatibility Mode Bit, TARGET Transmitter' Figure for details. 1h (R/W) = See the 'Backwards Compatibility Mode Bit, TARGET Transmitter' Figure for details.

20.7.2.13 I2CPSC Register (Offset = Ch) [Reset = 0000h]

I2CPSC is shown in Figure 20-34 and described in Table 20-23.

Return to the Summary Table.

The I2C prescaler register (I2CPSC) is a 16-bit register (see Figure 14-21) used for dividing down the I2C input clock to obtain the desired module clock for the operation of the I2C module. See the device-specific data manual for the supported range of values for the module clock frequency.
IPSC must be initialized while the I2C module is in reset (IRS = 0 in I2CMDR). The prescaled frequency takes effect only when IRS is changed to 1. Changing the IPSC value while IRS = 1 has no effect.

Figure 20-34. I2CPSC Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
IPSC							
R/W-0h							

Table 20-23. I2CPSC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	IPSC	R/W	0h	I2C prescaler divide-down value. IPSC determines how much the CPU clock is divided to create the module clock of the I2C module: module clock frequency = I2C input clock frequency/(IPSC + 1) Note: IPSC must be initialized while the I2C module is in reset (IRS = 0 in I2CMDR). Reset type: SYSRSn

20.7.2.14 I2CFFTX Register (Offset = 20h) [Reset = 0000h]

I2CFFTX is shown in Figure 20-35 and described in Table 20-24.

Return to the Summary Table.

The I2C transmit FIFO register (I2CFFTX) is a 16-bit register that contains the I2C FIFO mode enable bit as well as the control and status bits for the transmit FIFO mode of operation on the I2C peripheral.

Figure 20-35. I2CFFTX Register

15	14	13	12	11	10	9	8
RESERVED	I2CFFEN	TXFFRST	TXFFST				
R-0h	R/W-0h	R/W-0h	R-0h				
7	6	5	4	3	2	1	0
TXFFINT	TXFFINTCLR	TXFFIENA	TXFFIL				
R-0h	R-0/W1S-0h	R/W-0h	R/W-0h				

Table 20-24. I2CFFTX Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14	I2CFFEN	R/W	0h	I2C FIFO mode enable bit. This bit must be enabled for either the transmit or the receive FIFO to operate correctly. Reset type: SYSRSn 0h (R/W) = Disable the I2C FIFO mode. 1h (R/W) = Enable the I2C FIFO mode.
13	TXFFRST	R/W	0h	Transmit FIFO Reset Reset type: SYSRSn 0h (R/W) = Reset the transmit FIFO pointer to 0000 and hold the transmit FIFO in the reset state. 1h (R/W) = Enable the transmit FIFO operation.
12-8	TXFFST	R	0h	Contains the status of the transmit FIFO: xxxxx Transmit FIFO contains xxxxx bytes. 00000 Transmit FIFO is empty. Note: Since these bits are reset to zero, the transmit FIFO interrupt flag will be set when the transmit FIFO operation is enabled and the I2C is taken out of reset. This will generate a transmit FIFO interrupt if enabled. To avoid any detrimental effects from this, write a one to the TXFFINTCLR once the transmit FIFO operation is enabled and the I2C is taken out of reset. Reset type: SYSRSn
7	TXFFINT	R	0h	Transmit FIFO interrupt flag. This bit cleared by a CPU write of a 1 to the TXFFINTCLR bit. If the TXFFIENA bit is set, this bit will generate an interrupt when it is set. Reset type: SYSRSn 0h (R/W) = Transmit FIFO interrupt condition has not occurred. 1h (R/W) = Transmit FIFO interrupt condition has occurred.
6	TXFFINTCLR	R-0/W1S	0h	Transmit FIFO Interrupt Flag Clear Reset type: SYSRSn 0h (R/W) = Writes of zeros have no effect. Reads return a 0. 1h (R/W) = Writing a 1 to this bit clears the TXFFINT flag.
5	TXFFIENA	R/W	0h	Transmit FIFO Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disabled. TXFFINT flag does not generate an interrupt when set. 1h (R/W) = Enabled. TXFFINT flag does generate an interrupt when set.

Table 20-24. I2CFFTX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	TXFFIL	R/W	0h	Transmit FIFO interrupt level. These bits set the status level that will set the transmit interrupt flag. When the TXFFST4-0 bits reach a value equal to or less than these bits, the TXFFINT flag will be set. This will generate an interrupt if the TXFFIENA bit is set. Because the I2C on this device has a 16-level transmit FIFO, these bits cannot be configured for an interrupt of more than 16 FIFO levels. Reset type: SYSRSn

20.7.2.15 I2CFFRX Register (Offset = 21h) [Reset = 0000h]

I2CFFRX is shown in Figure 20-36 and described in Table 20-25.

Return to the Summary Table.

The I2C receive FIFO register (I2CFFRX) is a 16-bit register that contains the control and status bits for the receive FIFO mode of operation on the I2C peripheral.

Figure 20-36. I2CFFRX Register

15	14	13	12	11	10	9	8
RESERVED		RXFFRST	RXFFST				
R-0h		R/W-0h	R-0h				
7	6	5	4	3	2	1	0
RXFFINT	RXFFINTCLR	RXFFIENA	RXFFIL				
R-0h	R-0/W1S-0h	R/W-0h	R/W-0h				

Table 20-25. I2CFFRX Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13	RXFFRST	R/W	0h	I2C receive FIFO reset bit Reset type: SYSRSn 0h (R/W) = Reset the receive FIFO pointer to 0000 and hold the receive FIFO in the reset state. 1h (R/W) = Enable the receive FIFO operation.
12-8	RXFFST	R	0h	Contains the status of the receive FIFO: xxxxx Receive FIFO contains xxxxx bytes 00000 Receive FIFO is empty. Reset type: SYSRSn
7	RXFFINT	R	0h	Receive FIFO interrupt flag. This bit cleared by a CPU write of a 1 to the RXFFINTCLR bit. If the RXFFIENA bit is set, this bit will generate an interrupt when it is set Reset type: SYSRSn 0h (R/W) = Receive FIFO interrupt condition has not occurred. 1h (R/W) = Receive FIFO interrupt condition has occurred.
6	RXFFINTCLR	R-0/W1S	0h	Receive FIFO interrupt flag clear bit. Reset type: SYSRSn 0h (R/W) = Writes of zeros have no effect. Reads return a zero. 1h (R/W) = Writing a 1 to this bit clears the RXFFINT flag.
5	RXFFIENA	R/W	0h	Receive FIFO interrupt enable bit. Reset type: SYSRSn 0h (R/W) = Disabled. RXFFINT flag does not generate an interrupt when set. 1h (R/W) = Enabled. RXFFINT flag does generate an interrupt when set.
4-0	RXFFIL	R/W	0h	Receive FIFO interrupt level. These bits set the status level that will set the receive interrupt flag. When the RXFFST4-0 bits reach a value equal to or greater than these bits, the RXFFINT flag is set. This will generate an interrupt if the RXFFIENA bit is set. Note: Since these bits are reset to zero, the receive FIFO interrupt flag will be set if the receive FIFO operation is enabled and the I2C is taken out of reset. This will generate a receive FIFO interrupt if enabled. To avoid this, modify these bits on the same instruction as or prior to setting the RXFFRST bit. Because the I2C on this device has a 16-level receive FIFO, these bits cannot be configured for an interrupt of more than 16 FIFO levels. Reset type: SYSRSn

Chapter 21

Serial Communications Interface (SCI)

This chapter describes the features and operation of the serial communication interface (SCI) module. SCI is a two-wire asynchronous serial port, commonly known as a UART. The SCI modules support digital communications between the CPU and other asynchronous peripherals that use the standard non-return-to-zero (NRZ) format. The SCI receiver and transmitter each have a 16-level deep FIFO for reducing servicing overhead, and each has a separate enable and interrupt bits. Both can be operated independently for half-duplex communication, or simultaneously for full-duplex communication.

To specify data integrity, the SCI checks received data for break detection, parity, overrun, and framing errors. The bit rate is programmable to different speeds through a 16-bit baud-select register.

21.1 Introduction

The SCI interfaces are shown in Figure 21-1.

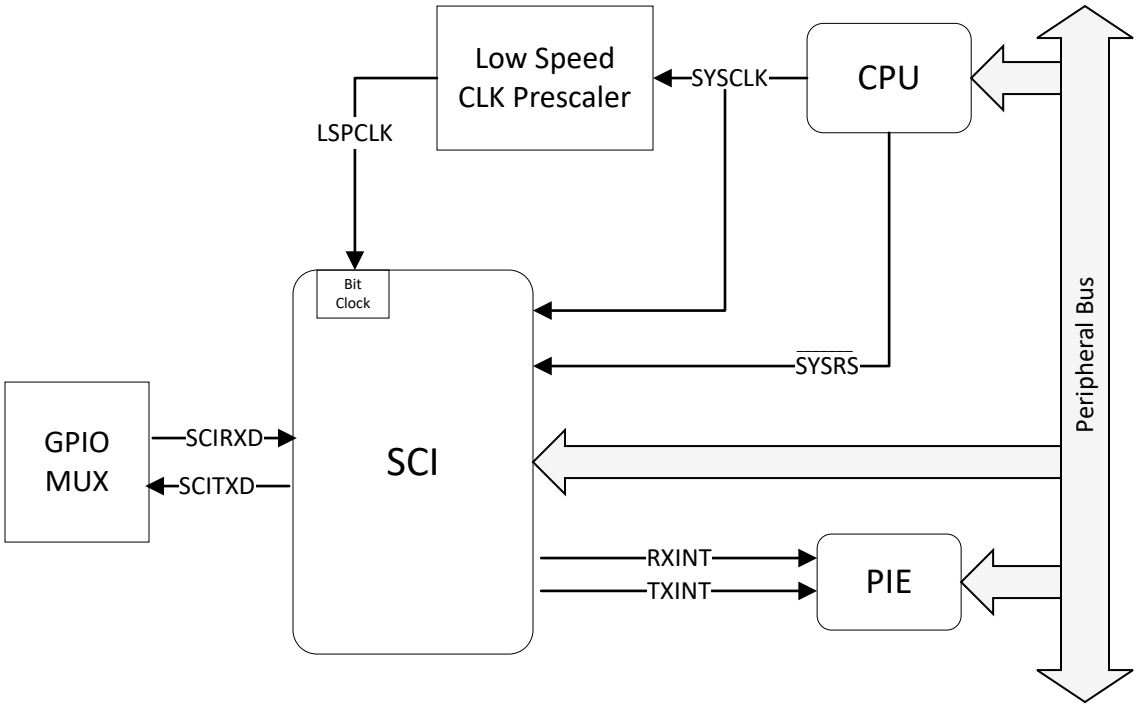


Figure 21-1. SCI CPU Interface

21.1.1 Features

Features of the SCI module include:

- Two external pins (both pins can be used as GPIO, if not used for SCI):
  - SCITXD: SCI transmit-output pin
  - SCIRXD: SCI receive-input pin
- Baud rate programmable to 64K different rates
- Data-word format
  - One start bit
  - Data-word length programmable from one to eight bits
  - Optional even/odd/no parity bit
  - One or two stop bits
  - An extra bit to distinguish addresses from data (address bit mode only)
- Four error-detection flags: parity, overrun, framing, and break detection
- Two wake-up multiprocessor modes: idle-line and address bit
- Half- or full-duplex operation
- Double-buffered receive and transmit functions
- Transmitter and receiver operations can be accomplished through interrupt-driven or polled algorithms with status flags
- Separate enable bits for transmitter and receiver interrupts (except BRKDT)
- NRZ (non-return-to-zero) format

Enhanced features include:

- Auto-baud-detect hardware logic
- 16-level transmit/receive FIFO

21.1.2 SCI Related Collateral

Foundational Materials

- C28x Academy - SCI
- One Minute RS-485 Introduction (Video)
- RS-232, RS-422, RS-485: What Are the Differences? (Video)

Getting Started Materials

- [FAQ] My C2000 SCI is not Transmitting and/or Receiving data correctly, how do I fix this?

21.1.3 Block Diagram

Figure 21-2 shows the SCI module block diagram. The SCI port operation is configured and controlled by the registers listed in Section 21.15.

21.2 Architecture

The major elements used in full-duplex operation are shown in Figure 21-2 and include:

- A transmitter (TX) and the major registers (upper half of Figure 21-2):
  - SCITXBUF — transmitter data buffer register. Contains data (loaded by the CPU) to be transmitted
  - TXSHF register — transmitter shift register. Accepts data from register SCITXBUF and shifts data onto the SCITXD pin, one bit at a time
- A receiver (RX) and the major registers (lower half of Figure 21-2):
  - RXSHF register — receiver shift register. Shifts data in from SCIRXD pin, one bit at a time
  - SCIRXBUF — receiver data buffer register. Contains data to be read by the CPU. Data from a remote processor is loaded into register RXSHF and then into registers SCIRXBUF and SCIRXEMU
- A programmable baud generator
- Control and status registers

The SCI receiver and transmitter can operate either independently or simultaneously.

21.3 SCI Module Signal Summary

A summarized description of each SCI signal name is shown in Table 21-1.

Table 21-1. SCI Module Signal Summary

Signal Name	Description
External signals	
SCIRXD	SCI Asynchronous Serial Port receive data
SCITXD	SCI Asynchronous Serial Port transmit data
Control	
Baud clock	LSPCLK Prescaled clock
Interrupt signals	
TXINT	Transmit interrupt
RXINT	Receive interrupt

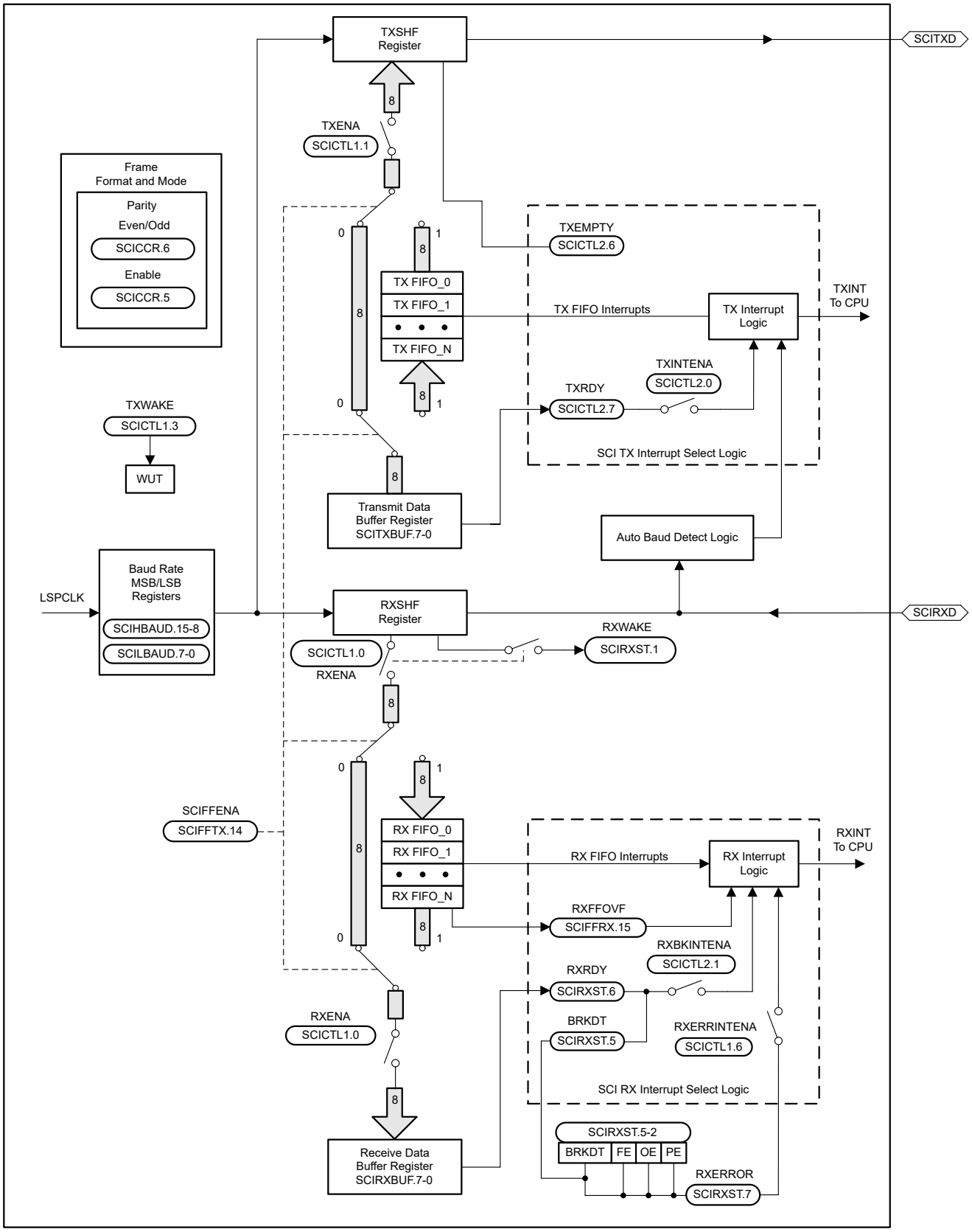


Figure 21-2. Serial Communications Interface (SCI) Module Block Diagram

21.4 Configuring Device Pins

The GPIO mux registers must be configured to connect this peripheral to the device pins. To avoid glitches on the pins, the GPyGMUX bits must be configured first (while keeping the corresponding GPyMUX bits at the default of zero), followed by writing the GPyMUX register to the desired value.

Some IO functionality is defined by GPIO register settings independent of this peripheral. For input signals, the GPIO input qualification must be set to asynchronous mode by setting the appropriate GPxQSELn register bits to 11b. The internal pullups can be configured in the GPyPUD register.

See the *General-Purpose Input/Output (GPIO)* chapter for more details on GPIO mux and settings.

21.5 Multiprocessor and Asynchronous Communication Modes

The SCI has two multiprocessor protocols, the idle-line multiprocessor mode (see Section 21.8) and the address-bit multiprocessor mode (see Section 21.9). These protocols allow efficient data transfer between multiple processors.

The SCI offers the universal asynchronous receiver/transmitter (UART) communications mode for interfacing with many popular peripherals. The asynchronous mode (see Section 21.10) requires two lines to interface with many standard devices such as terminals and printers that use RS-232-C formats. Data transmission characteristics include:

- One start bit
- One to eight data bits
- An even/odd parity bit or no parity bit
- One or two stop bits

21.6 SCI Programmable Data Format

SCI data, both receive and transmit, is in NRZ (non-return-to-zero) format. The NRZ data format, shown in Figure 21-3, consists of:

- One start bit
- One to eight data bits
- An even/odd parity bit (optional)
- One or two stop bits
- An extra bit to distinguish addresses from data (address-bit mode only)

The basic unit of data is called a character and is one to eight bits in length. Each character of data is formatted with a start bit, one or two stop bits, and optional parity and address bits. A character of data with formatting information is called a frame and is shown in Figure 21-3.

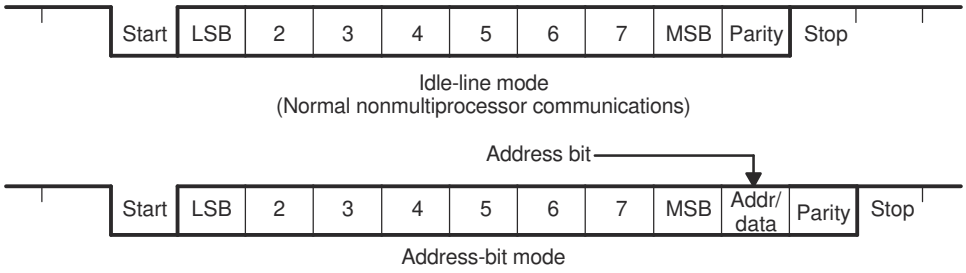


Figure 21-3. Typical SCI Data Frame Formats

To program the data format, use the SCICCR register. The bits used to program the data format are shown in Table 21-2.

Table 21-2. Programming the Data Format Using SCICCR

Bits	Bit Name	Designation	Functions
2-0	SCICHAR	SCICCR.2:0	Select the character (data) length (one to eight bits).
5	PARITYENA (ENABLE)	SCICCR.5	Enables the parity function if set to 1, or disables the parity function if cleared to 0.
6	PARITY (EVEN/ODD)	SCICCR.6	If parity is enabled, selects odd parity if cleared to 0; even parity if set to 1.
7	STOPBITS	SCICCR.7	Determines the number of stop bits transmitted—one stop bit if cleared to 0 or two stop bits if set to 1.

21.7 SCI Multiprocessor Communication

The multiprocessor communication format allows one processor to efficiently send blocks of data to other processors on the same serial link. On one serial line, there can be only one transfer at a time. In other words, there can be only one talker on a serial line at a time.

Address Byte

The first byte of a block of information that the talker sends contains an address byte that is read by all listeners. Only listeners with the correct address can be interrupted by the data bytes that follow the address byte. The listeners with an incorrect address remain uninterrupted until the next address byte.

Sleep Bit

All processors on the serial link set the SCI SLEEP bit (bit 2 of SCICTL1) to 1 so that the processor is interrupted only when the address byte is detected. When the processor reads a block address that corresponds to the CPU device address as set by your application software, your program must clear the SLEEP bit to enable the SCI to generate an interrupt on receipt of each data byte.

Although the receiver still operates when the SLEEP bit is 1, the receiver does not set RXRDY, RXINT, or any of the receiver error status bits to 1 unless the address byte is detected and the address bit in the received frame is a 1 (applicable to address-bit mode). The SCI does not alter the SLEEP bit; your software must alter the SLEEP bit.

21.7.1 Recognizing the Address Byte

A processor recognizes an address byte differently, depending on the multiprocessor mode used. For example:

- The idle-line mode (Section 21.8) leaves a quiet space before the address byte. This mode does not have an extra address/data bit and is more efficient than the address-bit mode for handling blocks that contain more than 10 bytes of data. The idle-line mode must be used for typical non-multiprocessor SCI communication.
- The address-bit mode (Section 21.9) adds an extra bit (that is, an address bit) into every byte to distinguish addresses from data. This mode is more efficient in handling many small blocks of data because, unlike the idle mode, this mode does not wait between blocks of data. However, at a high transmit speed, the program is not fast enough to avoid a 10-bit idle in the transmission stream.

21.7.2 Controlling the SCI TX and RX Features

The multiprocessor mode is software selectable using the ADDR/IDLE MODE bit (SCICCR, bit 3). Both modes use the TXWAKE flag bit (SCICTL1, bit 3), RXWAKE flag bit (SCIRXST, bit1), and the SLEEP flag bit (SCICTL1, bit 2) to control the SCI transmitter and receiver features of these modes.

21.7.3 Receipt Sequence

In both multiprocessor modes, the receive sequence is as follows:

1. At the receipt of an address block, the SCI port wakes up and requests an interrupt (bit number 1 RX/BK INT ENA-of SCICTL2 must be enabled to request an interrupt in non-FIFO mode of operation. In FIFO mode, RXFFINT serves this purpose and to enable this, RXFFINTEN in SCIFFRX register must be enabled with RXFFIL in the same register set to 1). The SCI reads the first frame of the block, which contains the destination address.
2. A software routine is entered through the interrupt and checks the incoming address. This address byte is checked against the device address byte stored in memory.
3. If the check shows that the block is addressed to the device CPU, the CPU clears the SLEEP bit and reads the rest of the block. If not, the software routine exits with the SLEEP bit still set, and does not receive interrupts until the next block start.

21.8 Idle-Line Multiprocessor Mode

In the idle-line multiprocessor protocol (ADDR/IDLE MODE bit = 0), blocks are separated by having a longer idle time between the blocks than between frames in the blocks. An idle time of ten or more high-level bits after a frame indicates the start of a new block. The time of a single bit is calculated directly from the baud value (bits per second). The idle-line multiprocessor communication format is shown in Figure 21-4 (ADDR/IDLE MODE bit is bit 3 of SCICCR).

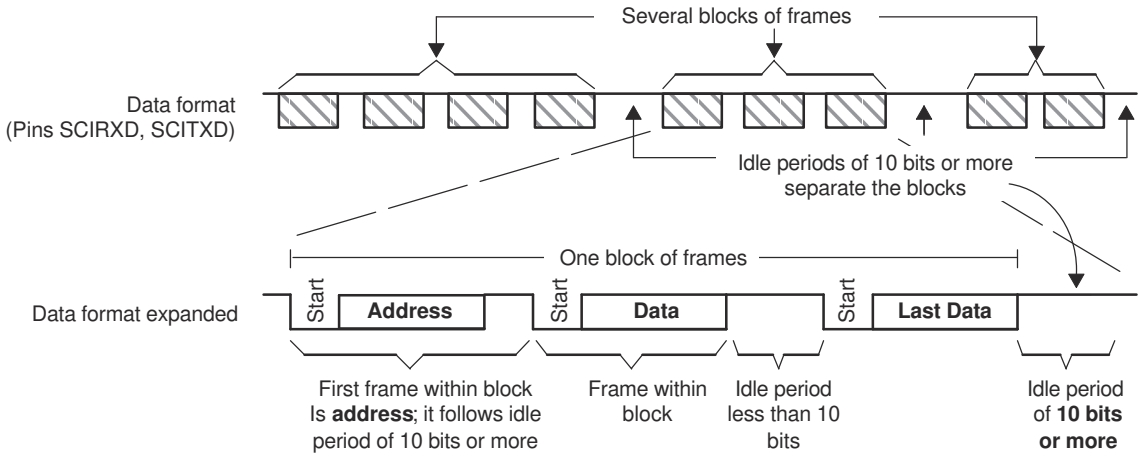


Figure 21-4. Idle-Line Multiprocessor Communication Format

21.8.1 Idle-Line Mode Steps

The steps followed by the idle-line mode:

1. SCI wakes up after receipt of the block-start signal.
2. The processor recognizes the next SCI interrupt.
3. The interrupt service routine compares the received address (sent by a remote transmitter) to the ISR address.
4. If the CPU is being addressed, the service routine clears the SLEEP bit and receives the rest of the data block.
5. If the CPU is not being addressed, the SLEEP bit remains set. This lets the CPU continue to execute the main program without being interrupted by the SCI port until the next detection of a block start.

Note

In IDLE mode, if the SCI is taking greater than 10 bit periods to read all the RXDATA from the FIFO, the SCI can miss the immediate block start to be detected.

The RXWAKE logic asserts only one time when the SCI identifies 10 bit periods of IDLE. The SCI does not assert again if RXBUF is read (which clears the WAKE condition) even if the line continues to be idle after RXBUF read.

So, if the ISR is taking more than 10 bit periods of time to read all the RXDATA from the FIFO using RXBUF, the SCI can miss to detect the next block start. This is applicable for both FIFO and Non-FIFO mode when the CPU takes greater than 10 bit clocks of SCI to read the data from RXBUF/ FIFO.

To avoid this, either of the following is recommended:

- Set SCICTL1.SWRESET after reading all RX data at the end of the ISR.
- Read and check the RXWAKE status bit before reading the RXBUF register. If RXWAKE is set, do not set the SLEEP bit for RX at the end of the ISR.

21.8.2 Block Start Signal

There are two ways to send a block-start signal:

- **Method 1:** Deliberately leave an idle time of ten bits or more by delaying the time between the transmission of the last frame of data in the previous block and the transmission of the address frame of the new block.
- **Method 2:** The SCI port first sets the TXWAKE bit (SCICTL1, bit 3) to 1 before writing to the SCITXBUF register. This sends an idle time of exactly 11 bits. In this method, the serial communications line is not idle any longer than necessary. (A don't care byte has to be written to SCITXBUF after setting TXWAKE, and before sending the address, so as to transmit the idle time.)

21.8.3 Wake-Up Temporary (WUT) Flag

Associated with the TXWAKE bit is the wake-up temporary (WUT) flag. WUT is an internal flag, double-buffered with TXWAKE. When TXSHF is loaded from SCITXBUF, WUT is loaded from TXWAKE, and the TXWAKE bit is cleared to 0. This arrangement is shown in Figure 21-5.

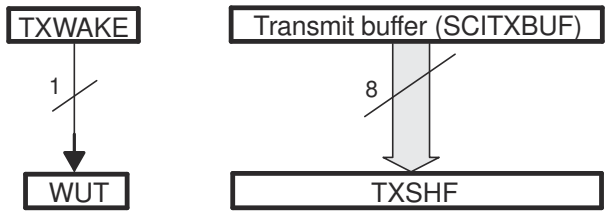


Figure 21-5. Double-Buffered WUT and TXSHF

21.8.3.1 Sending a Block Start Signal

To send out a block-start signal of exactly one frame time during a sequence of block transmissions:

1. Write a 1 to the TXWAKE bit.
2. Write a data word (content not important: a don’t care) to the SCITXBUF register (transmit data buffer) to send a block-start signal. (The first data word written is suppressed while the block-start signal is sent out and ignored after that.) When the TXSHF (transmit shift register) is free again, SCITXBUF contents are shifted to TXSHF, the TXWAKE value is shifted to WUT, and then TXWAKE is cleared.

   Because TXWAKE was set to a 1, the start, data, and parity bits are replaced by an idle period of 11 bits transmitted following the last stop bit of the previous frame.
3. Write a new address value to SCITXBUF.

   A don’t-care data word must first be written to register SCITXBUF so that the TXWAKE bit value can be shifted to WUT. After the don’t-care data word is shifted to the TXSHF register, the SCITXBUF (and TXWAKE, if necessary) can be written to again because TXSHF and WUT are both double-buffered.

21.8.4 Receiver Operation

The receiver operates regardless of the SLEEP bit. However, the receiver neither sets RXRDY nor the error status bits, nor does the receiver request a receive interrupt until an address frame is detected.

21.9 Address-Bit Multiprocessor Mode

In the address-bit protocol (ADDR/IDLE MODE bit = 1), frames have an extra bit called an address bit that immediately follows the last data bit. The address bit is set to 1 in the first frame of the block and to 0 in all other frames. The idle period timing is irrelevant (see Figure 21-6).

21.9.1 Sending an Address

The TXWAKE bit value is placed in the address bit. During transmission, when the SCITXBUF register and TXWAKE are loaded into the TXSHF register and WUT respectively, TXWAKE is reset to 0 and WUT becomes the value of the address bit of the current frame. Thus, to send an address:

1. Set the TXWAKE bit to 1 and write the appropriate address value to the SCITXBUF register.

   When this address value is transferred to the TXSHF register and shifted out, the address bit is sent as a 1. This flags the other processors on the serial link to read the address.
2. Write to SCITXBUF and TXWAKE after TXSHF and WUT are loaded. (Can be written to immediately since both TXSHF and WUT are both double-buffered.
3. Leave the TXWAKE bit set to 0 to transmit non-address frames in the block.

Note

As a general rule, the address-bit format is typically used for data frames of 11 bytes or less. This format adds one bit value (1 for an address frame, 0 for a data frame) to all data bytes transmitted. The idle-line format is typically used for data frames of 12 bytes or more.

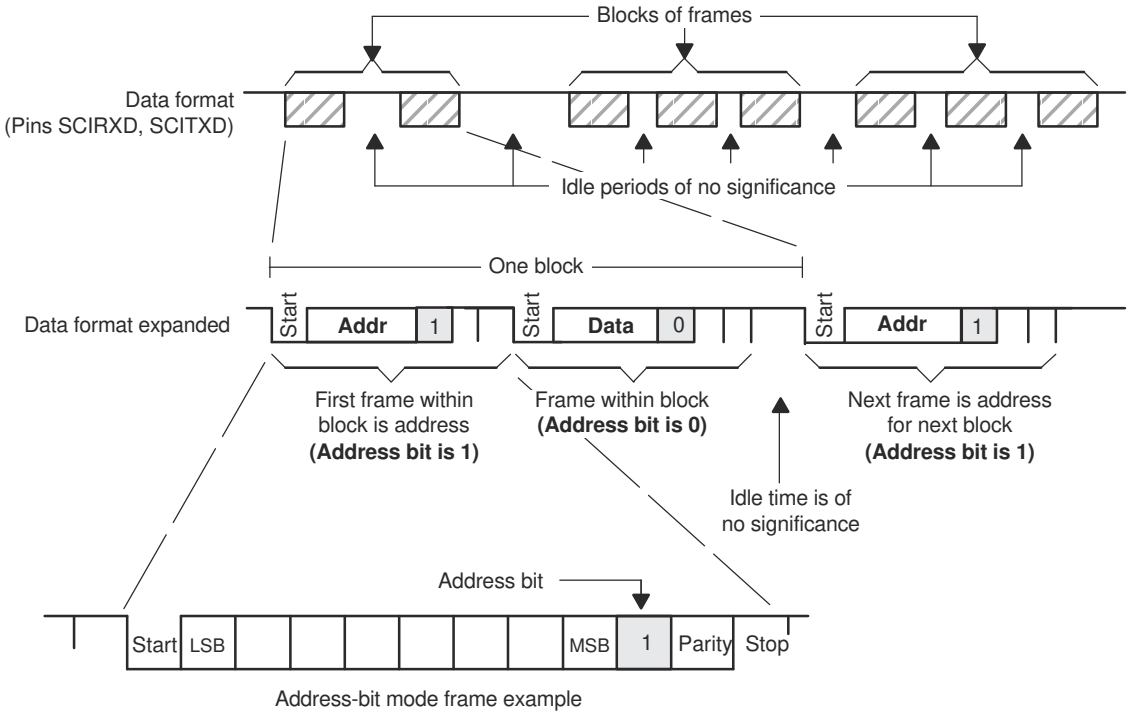


Figure 21-6. Address-Bit Multiprocessor Communication Format

21.10 SCI Communication Format

The SCI asynchronous communication format uses either single line (one way) or two line (two way) communications. In this mode, the frame consists of a start bit, one to eight data bits, an optional even/odd parity bit, and one or two stop bits (shown in Figure 21-7). There are eight SCICLK periods per data bit.

The receiver begins operation on receipt of a valid start bit. A valid start bit is identified by four consecutive internal SCICLK periods of zero bits as shown in Figure 21-7. If any bit is not zero, then the processor starts over and begins looking for another start bit.

For the bits following the start bit, the processor determines the bit value by making three samples in the middle of the bits. These samples occur on the fourth, fifth, and sixth SCICLK periods, and bit-value determination is on a majority (two out of three) basis. Figure 21-7 illustrates the asynchronous communication format for this with a start bit showing where a majority vote is taken.

Since the receiver synchronizes to frames, the external transmitting and receiving devices do not use a synchronized serial clock. The clock can be generated locally.

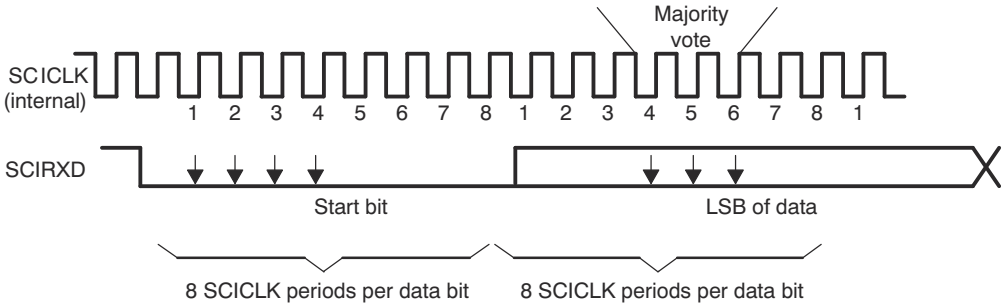


Figure 21-7. SCI Asynchronous Communications Format

21.10.1 Receiver Signals in Communication Modes

Figure 21-8 illustrates an example of receiver signal timing that assumes the following conditions:

- Address-bit wake-up mode (address bit does not appear in idle-line mode)
- Six bits per character

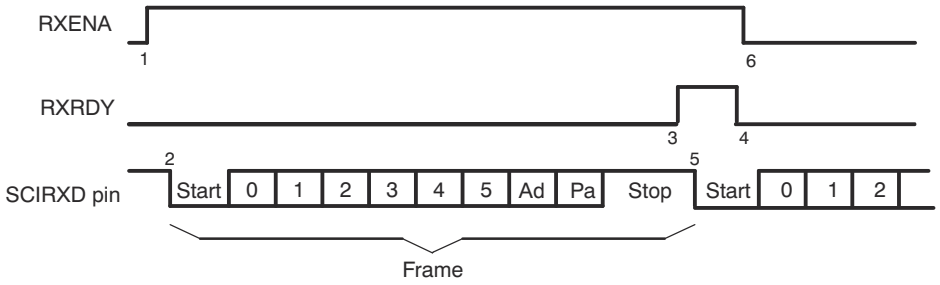


Figure 21-8. SCI RX Signals in Communication Modes

Notes:

1. Flag bit RXENA (SCICTL1, bit 0) goes high to enable the receiver.
2. Data arrives on the SCIRXD pin, start bit detected.
3. Data is shifted from RXSHF to the receiver buffer register (SCIRXBUF); an interrupt is requested. Flag bit RXRDY (SCIRXST, bit 6) goes high to signal that a new character has been received.
4. The program reads SCIRXBUF; flag RXRDY is automatically cleared.
5. The next byte of data arrives on the SCIRXD pin; the start bit is detected, then cleared.
6. Bit RXENA is brought low to disable the receiver. Data continues to be assembled in RXSHF but is not transferred to the receiver buffer register.

21.10.2 Transmitter Signals in Communication Modes

Figure 21-9 illustrates an example of transmitter signal timing that assumes the following conditions:

- Address-bit wake-up mode (address bit does not appear in idle-line mode)
- Three bits per character

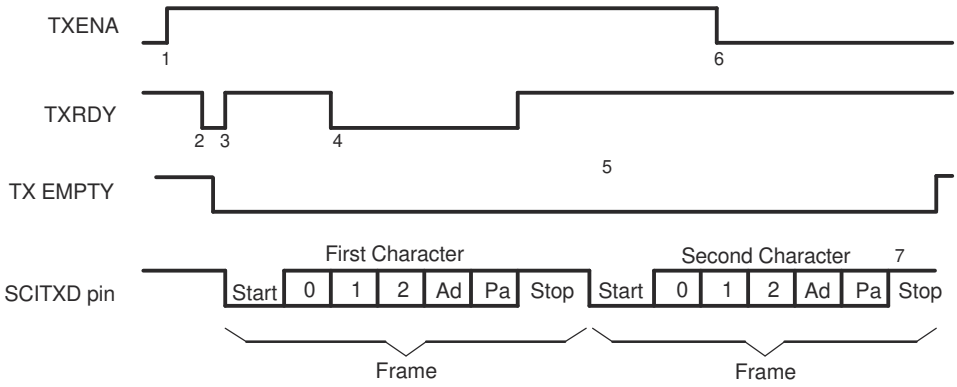


Figure 21-9. SCI TX Signals in Communications Mode

Notes:

1. Bit TXENA (SCICTL1, bit 1) goes high, enabling the transmitter to send data.
2. SCITXBUF is written to; thus, (1) the transmitter is no longer empty, and (2) TXRDY goes low.
3. The SCI transfers data to the shift register (TXSHF). The transmitter is ready for a second character (TXRDY goes high), and the transmitter requests an interrupt (to enable an interrupt, bit TX INT ENA — SCICTL2, bit 0 — must be set).
4. The program writes a second character to SCITXBUF after TXRDY goes high (item 3). (TXRDY goes low again after the second character is written to SCITXBUF.)
5. Transmission of the first character is complete. Transfer of the second character to shift register TXSHF begins.
6. Bit TXENA goes low to disable the transmitter; the SCI finishes transmitting the current character.
7. Transmission of the second character is complete; transmitter is empty and ready for new character.

21.11 SCI Port Interrupts

The SCI receiver and transmitter can be interrupt controlled. The SCICTL2 register has one flag bit (TXRDY) that indicates active interrupt conditions, and the SCIRXST register has two interrupt flag bits (RXRDY and BRKDT), plus the RX ERROR interrupt flag that is a logical-OR of the FE, OE, BRKDT, and PE conditions. The transmitter and receiver have separate interrupt-enable bits. When not enabled, the interrupts are not asserted; however, the condition flags remain active, reflecting transmission and receipt status.

The SCI has independent peripheral interrupt vectors for the receiver and transmitter. Peripheral interrupt requests can be either high priority or low priority. This is indicated by the priority bits that are output from the peripheral to the PIE controller. When both RX and TX interrupt requests are made at the same priority level, the receiver always has higher priority than the transmitter, reducing the possibility of receiver overrun.

The operation of peripheral interrupts is described in the Peripheral Interrupts section of the *System Control and Interrupts* chapter.

- If the RX/BK INT ENA bit (SCICTL2, bit 1) is set, the receiver peripheral interrupt request is asserted when one of the following events occurs:
  - The SCI receives a complete frame and transfers the data in the RXSHF register to the SCIRXBUF register. This action sets the RXRDY flag (SCIRXST, bit 6) and initiates an interrupt.
  - A break detect condition occurs (the SCIRXD is low for 9.625 bit periods following a missing stop bit). This action sets the BRKDT flag bit (SCIRXST, bit 5) and initiates an interrupt.
- If the TX INT ENA bit (SCICTL2.0) is set, the transmitter peripheral interrupt request is asserted whenever the data in the SCITXBUF register is transferred to the TXSHF register, indicating that the CPU can write to SCITXBUF; this action sets the TXRDY flag bit (SCICTL2, bit 7) and initiates an interrupt.

Note

Interrupt generation due to the RXRDY and BRKDT bits is controlled by the RX/BK INT ENA bit (SCICTL2, bit 1). Interrupt generation due to the RX ERROR bit is controlled by the RX ERR INT ENA bit (SCICTL1, bit 6).

21.11.1 Break Detect

A "break" signal (also called a "break detect" or "break sequence") can be sent to the module to signal to the bus a specific condition. This break signal is defined as a low pulse of a certain amount of time, typically at least 1 packet wide (including a missed stop bit). The SCI has two main methods for detecting a "break" signal sent on the line, with certain limitations for each.

The first for break detect method involves reading the SCIRXST.BRKDT bit. A break condition that triggers the BRKDT bit occurs when the SCI receiver data line (SCIRXD) remains continuously low for at least 9.625 bits, beginning after a missing first stop bit. If the SCIRX line goes high at any point during the 9.625 bits then the SCI does not flag a break detect. To trigger the first stop bit missed, the typical method is to hold the RX line low for:

- 1 start bit
- 8 data bits
- (optional) 1 address bit
- (optional) 1 parity bit
- 1 stop bit
- 9.625 bits of additional time held low

This is a total of 19.625 (systems using no parity or address bit), 20.625 (systems using either parity or address bit but not both), or 21.625 (systems using both parity and address bit) bit times held low.

The second method for break detect is to instead use the SCIRXST.FE, SCIRXST.PE and SCIRXBUF.SAR bits to detect a break signal of 10 or 11 bits of low. ISR code can use the following combination of flags and received data to determine if a break detect occurred:

- Break signal = 11 bits low (requires parity enabled)
  - FE == 1
  - PE == 1 for odd parity, PE == 0 for even parity
  - SCIRXBUF.SAR (received character) == 0x00
- Break signal = 10 bits low (requires parity disabled)
  - FE == 1
  - SCIRXBUF.SAR (received character) == 0x00

21.12 SCI Baud Rate Calculations

The internally generated serial clock is determined by the low-speed peripheral clock LSPCLK) and the baud-select registers. The SCI uses the 16-bit value of the baud-select registers to select one of the 64K different serial clock rates possible for a given LSPCLK.

See the bit descriptions in the baud-select registers, for the formula to use when calculating the SCI asynchronous baud. Table 21-3 shows the baud-select values for common SCI bit rates. LSPCLK/16 is the maximum baud rate. For example, if LSPCLK is 100MHz, then the maximum baud rate is 6.25Mbps.

Table 21-3. Asynchronous Baud Register Values for Common SCI Bit Rates

	LSPCLK Clock Frequency, 100MHz		
Baud Rate	BRR	Actual Baud Rate	% Error
2400	5207 (1457h)	2400	0
4800	2603 (A2Bh)	4800	0
9600	1301 (515h)	9601	0.01
19200	650 (28Ah)	19201	0.01
38400	324 (144h)	38462	0.16

21.13 SCI Enhanced Features

The C28x SCI features autobaud detection and transmit/receive FIFO. The following section explains the FIFO operation.

21.13.1 SCI FIFO Description

The following steps explain the FIFO features and help with programming the SCI with FIFOs.

1. **Reset.** At reset the SCI powers up in standard SCI mode and the FIFO function is disabled. The FIFO registers SCIFFTX, SCIFFRX, and SCIFFCT remain inactive.
2. **Standard SCI.** The standard SCI modes work normally with TXINT/RXINT interrupts as the interrupt source for the module.
3. **FIFO enable.** FIFO mode is enabled by setting the SCIFFEN bit in the SCIFFTX register. SCIRST can reset the FIFO mode at any stage of the operation.
4. **Active registers.** All the SCI registers and SCI FIFO registers (SCIFFTX, SCIFFRX, and SCIFFCT) are active.
5. **Interrupts.** FIFO mode has two interrupts; transmit FIFO (TXINT) and receive FIFO (RXINT). The RXINT is the common interrupt for SCI FIFO receive, receive error, and receive FIFO overflow conditions. The TXINT of the standard SCI is disabled and this interrupt serves as SCI transmit FIFO interrupt.
6. **Buffers.** Transmit and receive buffers are supplemented with two 16-level FIFOs. The transmit FIFO registers are 8-bits wide and receive FIFO registers are 10-bits wide. The one-word transmit buffer (SCITXBUF) of the standard SCI functions as a transition buffer before the transmit FIFO and shift register. SCITXBUF is loaded into either the FIFO (when FIFO is enabled) or the TXSHF (when FIFO is disabled). When FIFO is enabled, SCITXBUF loads into the FIFO only after the last bit of the shift register is shifted out, so SCITXBUF cannot be treated as an additional level of buffer. With the FIFO enabled, TXSHF is directly loaded from the FIFO (not TXBUF) after an optional delay value (SCIFFCT). When FIFO mode is enabled for SCI, characters written to SCITXBUF are queued in to SCI-TXFIFO and the characters received in SCI-RXFIFO can be read using SCIRXBUF.
7. **Delayed transfer.** The rate that words in the FIFO are transferred to the transmit shift register is programmable. The SCIFFCT register bits (7−0) FFTXDLY7−FFTXDLY0 define the delay between the word transfer. The delay is defined in the number SCI baud clock cycles. The 8 bit register can define a minimum delay of 0 baud clock cycles and a maximum of 256-baud clock cycles. With zero delay, the SCI module can transmit data in continuous mode with the FIFO words shifting out back to back. With the 256 clock delay the SCI module can transmit data in a maximum delayed mode with the FIFO words shifting out with a delay of 256 baud clocks between each words. The programmable delay facilitates communication with slow SCI/UARTs with little CPU intervention.
8. **FIFO status bits.** Both the transmit and receive FIFOs have status bits TXFFST or RXFFST (bits 12−8) that define the number of words available in the FIFOs at any time. The transmit FIFO reset bit TXFIFO and receive reset bit RXFIFO reset the FIFO pointers to zero when these bits are cleared to 0. The FIFOs resumes operation from start once these bits are set to 1.
9. **Programmable interrupt levels.** Both transmit and receive FIFO can generate CPU interrupts. The interrupt trigger is generated whenever the transmit FIFO status bits TXFFST (bits 12−8) match (less than or equal to) the interrupt trigger level bits TXFFIL (bits 4−0). This provides a programmable interrupt trigger for transmit and receive sections of the SCI. Default value for these trigger level bits is 0x11111 for receive FIFO and 0x00000 for transmit FIFO, respectively.

Figure 21-10 and Table 21-4 explain the operation/configuration of SCI interrupts in nonFIFO/FFO mode.

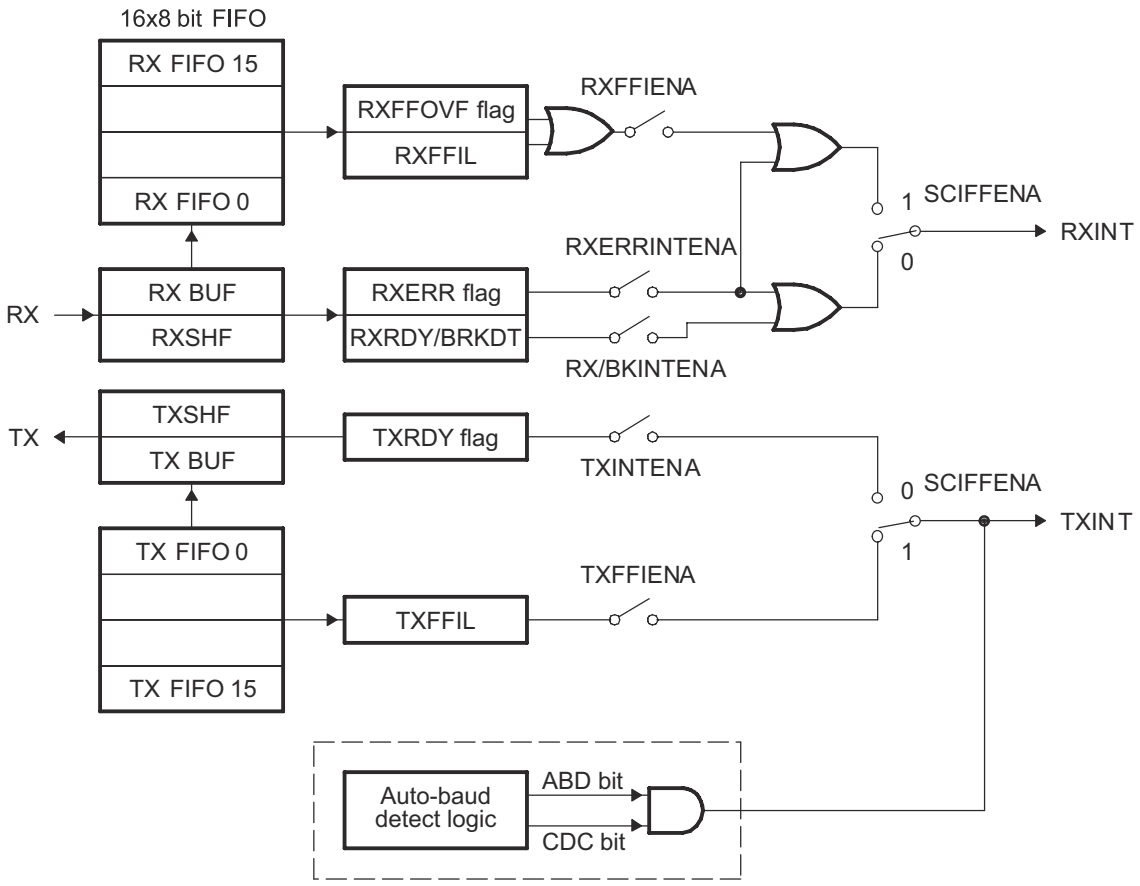


Figure 21-10. SCI FIFO Interrupt Flags and Enable Logic

Table 21-4. SCI Interrupt Flags

FIFO Options[1]	SCI Interrupt Source	Interrupt Flags	Interrupt Enables	FIFO Enable SCIFFENA	Interrupt Line
SCI without FIFO	Receive error	RXERR[2]	RXERRINTENA	0	RXINT
	Receive break	BRKDT	RX/BKINTENA	0	RXINT
	Data receive	RXRDY	RX/BKINTENA	0	RXINT
	Transmit empty	TXRDY	TXINTENA	0	TXINT
SCI with FIFO	Receive error and receive break	RXERR	RXERRINTENA	1	RXINT
	FIFO receive	RXFFIL	RXFFIENA	1	RXINT
	Transmit empty	TXFFIL	TXFFIENA	1	TXINT
Auto-baud	Auto-baud detected	ABD	Don't care	x	TXINT

(1) FIFO mode TXSHF is directly loaded after delay value, TXBUF is not used.
(2) RXERR can be set by BRKDT, FE, OE, PE flags. In FIFO mode, BRKDT interrupt is only through RXERR flag.

21.13.2 SCI Auto-Baud

Most SCI modules do not have an auto-baud detect logic built-in hardware. These SCI modules are integrated with embedded controllers whose clock rates are dependent on PLL reset values. Often embedded controller clocks change after final design. In the enhanced feature set this module supports an autobaud-detect logic in hardware. The following section explains the enabling sequence for autobaud-detect feature.

21.13.3 Autobaud-Detect Sequence

Bits ABD and CDC in SCIFFCT control the autobaud logic. The SCIRST bit can be enabled to make autobaud logic work.

If ABD is set while CDC is 1, which indicates auto-baud alignment, SCI transmit FIFO interrupt occurs (TXINT). After the interrupt service, the CDC bit must be cleared by software. If CDC remains set even after interrupt service, there can be no repeat interrupts.

1. Enable autobaud-detect mode for the SCI by setting the CDC bit (bit 13) in SCIFFCT and clearing the ABD bit (bit 15) by writing a 1 to ABDCLR bit (bit 14).
2. Initialize the baud register to be 1 or less than a baud rate limit of 500Kbps.
3. Allow SCI to receive either character "A" or "a" from a host at the desired baud rate. If the first character is either "A" or "a", the autobaud-detect hardware detects the incoming baud rate and sets the ABD bit.
4. The auto-detect hardware updates the baud rate register with the equivalent baud value hex. The logic also generates an interrupt to the CPU.
5. Respond to the interrupt clear ADB bit by writing a 1 to ABD CLR (bit 14) of SCIFFCT register and disable further autobaud locking by clearing CDC bit by writing a 0.
6. Read the receive buffer for character "A" or "a" to empty the buffer and buffer status.
7. If ABD is set while CDC is 1, which indicates autobaud alignment, the SCI transmit FIFO interrupt occurs (TXINT). After the interrupt service, the CDC bit must be cleared by software.

Note

At higher baud rates, the slew rate of the incoming data bits can be affected by transceiver and connector performance. While normal serial communications can work well, this slew rate can limit reliable autobaud detection at higher baud rates (typically beyond 100k baud) and cause the auto-baudlock feature to fail.

To avoid this, the following is recommended:

- Achieve a baud-lock between the host and C28x SCI boot loader using a lower baud rate.
- The host can then handshake with the loaded C28x application to set the SCI baud rate register to the desired higher baud rate.

21.14 Software

21.14.1 SCI Examples

NOTE: These examples are located in the C2000Ware installation at the following location:
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/sci

Cloud access to these examples is available at the following link: dev.ti.com C2000Ware Examples.

21.14.1.1 Tune Baud Rate via UART Example

FILE: baud_tune_via_uart.c

This example demonstrates the process of tuning the UART/SCI baud rate of a C2000 device based on the UART input from another device. As UART does not have a clock signal, reliable communication requires baud rates to be reasonably matched. This example addresses cases where a clock mismatch between devices is greater than is acceptable for communications, requiring baud compensation between boards. As reliable communication only requires matching the EFFECTIVE baud rate, it does not matter which of the two boards executes the tuning (the board with the less-accurate clock source does not need to be the one to tune; as long as one of the two devices tunes to the other, then proper communication can be established).

To tune the baud rate of this device, SCI data (of the desired baud rate) must be sent to this device.
The input SCI baud rate must be within the +/- MARGINPERCENT of the TARGETBAUD chosen below.
These two variables are defined below, and should be chosen based on the application requirements. Higher MARGINPERCENT will allow more data to be considered "correct" in noisy conditions, and may decrease accuracy. The TARGETBAUD is what was expected to be the baud rate, but due to clock differences, needs to be tuned for better communication robustness with the other device.

NOTE: Lower baud rates have more granularity in register options, and therefore tuning is more affective at these speeds.

This example project has support for migration across our C2000 device families. If you are wanting to build this project from launchpad or controlCARD, please specify in the .syscfg file the board you're using. At any time you can select another device to migrate this example. *External Connections* for Control Card

- SCIA_RX/eCAP1 is on GPIO0, connect to incoming SCI communications
- SCIA_TX is on GPIO1, for observation externally

Watch Variables

- *avgBaud* - Baud rate that was detected and set after tuning

21.14.1.2 SCI FIFO Digital Loop Back

FILE: sci_ex1_loopback.c

This program uses the internal loop back test mode of the peripheral. Other then boot mode pin configuration, no other hardware configuration is required. The pinmux and SCI modules are configured through the sysconfig file.

This test uses the loopback test mode of the SCI module to send characters starting with 0x00 through 0xFF. The test will send a character and then check the receive buffer for a correct match.

This example project has support for migration across our C2000 device families. If you are wanting to build this project from launchpad or controlCARD, please specify in the .syscfg file the board you're using. At any time you can select another device to migrate this example. *Watch Variables*

- *loopCount* - Number of characters sent
- *errorCount* - Number of errors detected
- *sendChar* - Character sent
- *receivedChar* - Character received

21.14.1.3 SCI Digital Loop Back with Interrupts

FILE: sci_ex2_loopback_interrupts.c

This test uses the internal loop back test mode of the peripheral. Other then boot mode pin configuration, no other hardware configuration is required. Both interrupts and the SCI FIFOs are used.

A stream of data is sent and then compared to the received stream. The SCI-A sent data looks like this:
00 01
01 02
02 03
....
FE FF
FF 00
etc..
The pattern is repeated forever.

Watch Variables
- *sDataA* - Data being sent
- *rDataA* - Data received
- *rDataPointA* - Keep track of where we are in the data stream. This is used to check the incoming data

21.14.1.4 SCI Echoback

FILE: sci_ex3_echoback.c

This test receives and echo-backs data through the SCI-A port.

A terminal such as 'putty' can be used to view the data from the SCI and to send information to the SCI. Characters received by the SCI port are sent back to the host.

Running the Application Open a COM port with the following settings using a terminal:
- Find correct COM port
- Bits per second = 9600
- Data Bits = 8
- Parity = None
- Stop Bits = 1
- Hardware Control = None

The program will print out a greeting and then ask you to enter a character which it will echo back to the terminal.

Watch Variables
- loopCounter - the number of characters sent

External Connections
Connect the USB cable from Control card J1:A to PC

21.14.1.5 stdout redirect example

FILE: sci_ex4_stdout_redirect.c This test transmits data through the SCI-A port to a terminal

A terminal such as 'putty' can be used to view the data from the SCI. Characters received by the SCI port are sent back to the host.

Running the Application Open a COM port with the following settings using a terminal:
- Find correct COM port
- Bits per second = 9600
- Data Bits = 8
- Parity = None
- Stop Bits = 1
- Hardware Control = None

The program will print out three sentences: one to the SCIA, one to CCS, and a final one to SCIA.

External Connections

Connect the SCI-A port to a PC via a transceiver and cable.

- DEVICE_GPIO_PIN_SCIRXDA is SCI_A-RXD (Connect to Pin3, PC-TX, of serial DB9 cable)
- DEVICE_GPIO_PIN_SCITXDA is SCI_A-TXD (Connect to Pin2, PC-RX, of serial DB9 cable)

21.15 SCI Registers

This Section describes the SCI Registers.

21.15.1 SCI Base Address Table

Table 21-5. SCI Base Address Table

Bit Field Name		DriverLib Name	Base Address	CPU1.DMA	Pipeline Protected
Instance	Structure				
SciaRegs	SCI_REGS	SCIA_BASE	0x0000_7200	-	YES
ScibRegs	SCI_REGS	SCIB_BASE	0x0000_7210	-	YES

21.15.2 SCI_REGS Registers

Table 21-6 lists the memory-mapped registers for the SCI_REGS registers. All register offset addresses not listed in Table 21-6 should be considered as reserved locations and the register contents should not be modified.

Table 21-6. SCI_REGS Registers

Offset	Acronym	Register Name	Write Protection
0h	SCICCR	Communications control register	
1h	SCICTL1	Control register 1	
2h	SCIHBAUD	Baud rate (high) register	
3h	SCILBAUD	Baud rate (low) register	
4h	SCICTL2	Control register 2	
5h	SCIRXST	Receive status register	
6h	SCIRXEMU	Receive emulation buffer register	
7h	SCIRXBUF	Receive data buffer	
9h	SCITXBUF	Transmit data buffer	
Ah	SCIFFTX	FIFO transmit register	
Bh	SCIFFRX	FIFO receive register	
Ch	SCIFFCT	FIFO control register	
Fh	SCIPRI	SCI priority control	

Complex bit access types are encoded to fit into small table cells. Table 21-7 shows the codes that are used for access types in this section.

Table 21-7. SCI_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

21.15.2.1 SCICCR Register (Offset = 0h) [Reset = 0000h]

SCICCR is shown in Figure 21-11 and described in Table 21-8.

Return to the Summary Table.

SCICCR defines the character format, protocol, and communications mode used by the SCI.

Figure 21-11. SCICCR Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
STOPBITS	PARITY	PARITYENA	LOOPBKENA	ADDRIDLE_MODE	SCICHAR		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		

Table 21-8. SCICCR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	STOPBITS	R/W	0h	SCI number of stop bits. This bit specifies the number of stop bits transmitted. The receiver checks for only one stop bit. Reset type: SYSRSn 0h (R/W) = One stop bit 1h (R/W) = Two stop bits
6	PARITY	R/W	0h	SCI parity odd/even selection. If the PARITY ENABLE bit (SCICCR, bit 5) is set, PARITY (bit 6) designates odd or even parity (odd or even number of bits with the value of 1 in both transmitted and received characters). Reset type: SYSRSn 0h (R/W) = Odd parity 1h (R/W) = Even parity
5	PARITYENA	R/W	0h	SCI parity enable. This bit enables or disables the parity function. If the SCI is in the addressbit multiprocessor mode (set using bit 3 of this register), the address bit is included in the parity calculation (if parity is enabled). For characters of less than eight bits, the remaining unused bits should be masked out of the parity calculation. Reset type: SYSRSn 0h (R/W) = Parity disabled no parity bit is generated during transmission or is expected during reception 1h (R/W) = Parity is enabled
4	LOOPBKENA	R/W	0h	Loop Back test mode enable. This bit enables the Loop Back test mode where the Tx pin is internally connected to the Rx pin. Reset type: SYSRSn 0h (R/W) = Loop Back test mode disabled 1h (R/W) = Loop Back test mode enabled

Table 21-8. SCICCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	ADDRIDLE_MODE	R/W	0h	SCI multiprocessor mode control bit. This bit selects one of the multiprocessor protocols.Multiprocessor communication is different from the other communication modes because it uses SLEEP and TXWAKE functions (bits SCICTL1, bit 2 and SCICTL1, bit 3, respectively). The idle-line mode is usually used for normal communications because the address-bit mode adds an extra bit to the frame. The idle-line mode does not add this extra bit and is compatible with RS-232 type communications. Reset type: SYSRSn 0h (R/W) = Idle-line mode protocol selected 1h (R/W) = Address-bit mode protocol selected
2-0	SCICHAR	R/W	0h	Character-length control bits 2-0. These bits select the SCI character length from one to eight bits. Characters of less than eight bits are right-justified in SCIRXBUF and SCIRXEMU and are padded with leading zeros in SCIRXBUF. SCITXBUF doesn't need to be padded with leading zeros. Reset type: SYSRSn 0h (R/W) = SCICHAR_LENGTH_1 1h (R/W) = SCICHAR_LENGTH_2 2h (R/W) = SCICHAR_LENGTH_3 3h (R/W) = SCICHAR_LENGTH_4 4h (R/W) = SCICHAR_LENGTH_5 5h (R/W) = SCICHAR_LENGTH_6 6h (R/W) = SCICHAR_LENGTH_7 7h (R/W) = SCICHAR_LENGTH_8

21.15.2.2 SCICTL1 Register (Offset = 1h) [Reset = 0000h]

SCICTL1 is shown in Figure 21-12 and described in Table 21-9.

Return to the Summary Table.

SCICTL1 controls the receiver/transmitter enable, TXWAKE and SLEEP functions, and the SCI software reset.

Figure 21-12. SCICTL1 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RXERRINTENA	SWRESET	RESERVED	TXWAKE	SLEEP	TXENA	RXENA
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-9. SCICTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	RXERRINTENA	R/W	0h	SCI receive error interrupt enable. Setting this bit enables an interrupt if the RX ERROR bit (SCIRXST, bit 7) becomes set because of errors occurring. Reset type: SYSRSn 0h (R/W) = Receive error interrupt disabled 1h (R/W) = Receive error interrupt enabled
5	SWRESET	R/W	0h	SCI software reset (active low). Writing a 0 to this bit initializes the SCI state machines and operating flags (registers SCICTL2 and SCIRXST) to the reset condition. This reset will not reset the FIFO pointers or flush out the data in TX/RX FIFO. If you need to clear the FIFO then perform SWRESET + TXFFINT + RXFFINT or refer to a channel reset SCIFFTX[SCIRST]. The SW RESET bit does not affect any of the configuration bits. All affected logic is held in the specified reset state until a 1 is written to SW RESET (the bit values following a reset are shown beneath each register diagram in this section). Thus, after a system reset, re-enable the SCI by writing a 1 to this bit. Clear this bit after a receiver break detect (BRKDT flag, bit SCIRXST, bit 5). SW RESET affects the operating flags of the SCI, but it neither affects the configuration bits nor restores the reset values. Once SW RESET is asserted, the flags are frozen until the bit is deasserted. The affected flags are as follows: Value After SW SCI Flag Register Bit RESET 1 TXRDY SCICTL2, bit 7 1 TX EMPTY SCICTL2, bit 6 0 RXWAKE SCIRXST, bit 1 0 PE SCIRXST, bit 2 0 OE SCIRXST, bit 3 0 FE SCIRXST, bit 4 0 BRKDT SCIRXST, bit 5 0 RXRDY SCIRXST, bit 6 0 RX ERROR SCIRXST, bit 7 Reset type: SYSRSn 0h (R/W) = Writing a 0 to this bit initializes the SCI state machines and operating flags (registers SCICTL2 and SCIRXST) to the reset condition. 1h (R/W) = After a system reset, re-enable the SCI by writing a 1 to this bit. There is no time requirement to meet before writing a one to this bit after writing a zero.
4	RESERVED	R	0h	Reserved

Table 21-9. SCICTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	TXWAKE	R/W	0h	SCI transmitter wake-up method select. The TXWAKE bit controls selection of the data-transmit feature, depending on which transmit mode (idle-line or address-bit) is specified at the ADDR/IDLE MODE bit (SCICCR, bit 3) Reset type: SYSRSn 0h (R/W) = Transmit feature is not selected. In idle-line mode: write a 1 to TXWAKE, then write data to register SCITXBUF to generate an idle period of 11 data bits In address-bit mode: write a 1 to TXWAKE, then write data to SCITXBUF to set the address bit for that frame to 1 1h (R/W) = Transmit feature selected is dependent on the mode, idle-line or address-bit: TXWAKE is not cleared by the SW RESET bit (SCICTL1, bit 5) it is cleared by a system reset or the transfer of TXWAKE to the WUT flag.
2	SLEEP	R/W	0h	SCI sleep. The TXWAKE bit controls selection of the data-transmit feature, depending on which transmit mode (idle-line or address-bit) is specified at the ADDR/IDLE MODE bit (SCICCR, bit 3). In a multiprocessor configuration, this bit controls the receiver sleep function. Clearing this bit brings the SCI out of the sleep mode. The receiver still operates when the SLEEP bit is set however, operation does not update the receiver buffer ready bit (SCIRXST, bit 6, RXRDY) or the error status bits (SCIRXST, bit 5-2: BRKDT, FE, OE, and PE) unless the address byte is detected. SLEEP is not cleared when the address byte is detected. Reset type: SYSRSn 0h (R/W) = Sleep mode disabled 1h (R/W) = Sleep mode enabled
1	TXENA	R/W	0h	SCI transmitter enable. Data is transmitted through the SCITXD pin only when TXENA is set. If reset, transmission is halted but only after all data previously written to SCITXBUF has been sent. Data written into SCITXBUF when TXENA is disabled will not be transmitted even if the TXENA is enabled later. Reset type: SYSRSn 0h (R/W) = Transmitter disabled 1h (R/W) = Transmitter enabled
0	RXENA	R/W	0h	SCI receiver enable. Data is received on the SCIRXD pin and is sent to the receiver shift register and then the receiver buffers. This bit enables or disables the receiver (transfer to the buffers). Clearing RXENA stops received characters from being transferred to the two receiver buffers and also stops the generation of receiver interrupts. However, this will not stop RX errors from triggering interrupts. To disable interrupts from RX errors use the RXERRINTENA bit. To stop propagation of the BRKDT interrupt use the RXBKINTENA bit. The receiver shift register can continue to assemble characters even while RXENA is cleared. Thus, if RXENA is set during the reception of a character, the complete character will be transferred into the receiver buffer registers, SCIRXEMU and SCIRXBUF. Reset type: SYSRSn 0h (R/W) = Prevent received characters from transfer into the SCIRXEMU and SCIRXBUF receiver buffers 1h (R/W) = Send received characters to SCIRXEMU and SCIRXBUF

21.15.2.3 SCIHBAUD Register (Offset = 2h) [Reset = 0000h]

SCIHBAUD is shown in Figure 21-13 and described in Table 21-10.

Return to the Summary Table.

The values in SCIHBAUD and SCILBAUD specify the baud rate for the SCI.

Figure 21-13. SCIHBAUD Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
BAUD							
R/W-0h							

Table 21-10. SCIHBAUD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	BAUD	R/W	0h	SCI 16-bit baud selection Registers SCIHBAUD (MSbyte). The internally-generated serial clock is determined by the low speed peripheral clock (LSPCLK) signal and the two baud-select registers. The SCI uses the 16-bit value of these registers to select one of 64K serial clock rates for the communication modes. BRR = (SCIHBAUD << 8) + (SCILBAUD) The SCI baud rate is calculated using the following equation: SCI Asynchronous Baud = LSPCLK / ((BRR + 1) *8) Alternatively, BRR = LSPCLK / (SCI Asynchronous Baud * 8) - 1 Note that the above formulas are applicable only when 0 < BRR < 65536. If BRR = 0, then SCI Asynchronous Baud = LSPCLK / 16 Where: BRR = the 16-bit value (in decimal) in the baud-select registers Reset type: SYSRSn

21.15.2.4 SCILBAUD Register (Offset = 3h) [Reset = 0000h]

SCILBAUD is shown in Figure 21-14 and described in Table 21-11.

Return to the Summary Table.

The values in SCIHBAUD and SCILBAUD specify the baud rate for the SCI.

Figure 21-14. SCILBAUD Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
BAUD							
R/W-0h							

Table 21-11. SCILBAUD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	BAUD	R/W	0h	See SCIHBAUD Detailed Description Reset type: SYSRSn

21.15.2.5 SCICTL2 Register (Offset = 4h) [Reset = 00C0h]

SCICTL2 is shown in Figure 21-15 and described in Table 21-12.

Return to the Summary Table.

SCICTL2 enables the receive-ready, break-detect, and transmit-ready interrupts as well as transmitter-ready and -empty flags.

Figure 21-15. SCICTL2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TXRDY	TXEMPTY	RESERVED				RXBKINTENA	TXINTENA
R-1h	R-1h	R-0h				R/W-0h	R/W-0h

Table 21-12. SCICTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	TXRDY	R	1h	Transmitter buffer register ready flag. When set, this bit indicates that the transmit data buffer register, SCITXBUF, is ready to receive another character. Writing data to the SCITXBUF automatically clears this bit. When set, this flag asserts a transmitter interrupt request if the interrupt-enable bit, TX INT ENA (SCICTL2.0), is also set. TXRDY is set to 1 by enabling the SW RESET bit (SCICTL1.5) or by a system reset. Reset type: SYSRSn 0h (R/W) = SCITXBUF is full 1h (R/W) = SCITXBUF is ready to receive the next character
6	TXEMPTY	R	1h	Transmitter empty flag. This flag's value indicates the contents of the transmitter's buffer register (SCITXBUF) and shift register (TXSHF). An active SW RESET (SCICTL1.5), or a system reset, sets this bit. This bit does not cause an interrupt request. Reset type: SYSRSn 0h (R/W) = Transmitter buffer or shift register or both are loaded with data 1h (R/W) = Transmitter buffer and shift registers are both empty
5-2	RESERVED	R	0h	Reserved
1	RXBKINTENA	R/W	0h	Receiver-buffer/break interrupt enable. This bit controls the interrupt request caused by either the RXRDY flag or the BRKDT flag (bits SCIRXST.6 and .5) being set. However, RX/BK INT ENA does not prevent the setting of these flags. Reset type: SYSRSn 0h (R/W) = Disable RXRDY/BRKDT interrupt 1h (R/W) = Enable RXRDY/BRKDT interrupt

Table 21-12. SCICTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	TXINTENA	R/W	0h	SCITXBUF-register interrupt enable. This bit controls the interrupt request caused by the setting of TXRDY flag bit (SCICTL2.7). However, it does not prevent the TXRDY flag from being set (which indicates SCITXBUF is ready to receive another character). 0 Disable TXRDY interrupt 1 Enable TXRDY interrupt. In non-FIFO mode, a dummy (or a valid) data has to be written to SCITXBUF for the first transmit interrupt to occur. This is the case when you enable the transmit interrupt for the first time and also when you re-enable (disable and then enable) the transmit interrupt. If TXINTENA is enabled after writing the data to SCITXBUF, it will not generate an interrupt. Reset type: SYSRSn 0h (R/W) = Disable TXRDY interrupt 1h (R/W) = Enable TXRDY interrupt

21.15.2.6 SCIRXST Register (Offset = 5h) [Reset = 0000h]

SCIRXST is shown in Figure 21-16 and described in Table 21-13.

Return to the Summary Table.

SCIRXST contains seven bits that are receiver status flags (two of which can generate interrupt requests). Each time a complete character is transferred to the receiver buffers (SCIRXEMU and SCIRXBUF), the status flags are updated.

Figure 21-16. SCIRXST Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RXERROR	RXRDY	BRKDT	FE	OE	PE	RXWAKE	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 21-13. SCIRXST Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	RXERROR	R	0h	SCI receiver error flag. The RX ERROR flag indicates that one of the error flags in the receiver status register is set. RX ERROR is a logical OR of the break detect, framing error, overrun, and parity error enable flags (bits 5-2: BRKDT, FE, OE, and PE). A 1 on this bit will cause an interrupt if the RX ERR INT ENA bit (SCICTL1.6) is set. This bit can be used for fast error-condition checking during the interrupt service routine. This error flag cannot be cleared directly it is cleared by an active SW RESET, channel reset (SCIRST), or by a system reset. Reset type: SYSRSn 0h (R/W) = No error flags set 1h (R/W) = Error flag(s) set
6	RXRDY	R	0h	SCI receiver-ready flag. When a new character is ready to be read from the SCIRXBUF register, the receiver sets this bit, and a receiver interrupt is generated if the RX/BK INT ENA bit (SCICTL2.1) is a 1. RXRDY is cleared by a reading of the SCIRXBUF register, by an active SW RESET, channel reset (SCIRST), or by a system reset. Reset type: SYSRSn 0h (R/W) = No new character in SCIRXBUF 1h (R/W) = Character ready to be read from SCIRXBUF

Table 21-13. SCIRXST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	BRKDT	R	0h	SCI break-detect flag. The SCI sets this bit when a break condition occurs. A break condition occurs when the SCI receiver data line (SCIRXD) remains continuously low for at least 9.625 bits, beginning after a missing first stop bit. If the SCIRX line goes high at any point during the 9.625 bits then the SCI will not flag a break detect. In order to trigger the first stop bit missed, the typical method is to hold the RX line low for 1 start bit, 8 data bits, 1 optional address bit, 1 optional parity bit, 1 stop bit, and 9.625 bits of additional time held low. This is a total of 19.625 (no parity/address bit), 20.625 (either parity or address bit), or 21.625 (both parity and address bit) bit times. To instead detect a 'break seq' or 'break sequence' of 11 bits of low voltage level (0), ISR code can use the following combination of flags and received data: FE==1 && PE==1 && SCIRXBUF.SAR (received character)==0x00. This assumes parity enabled and odd parity set. With even parity, PE==0 instead. The detection of 11 bits of low/0 can be reduced to 10 bits of low if no parity bit is used (then PE flag does not matter to detect the sequence). The occurrence of a break causes a receiver interrupt to be generated if the RX/BK INT ENA bit is a 1, but it does not cause the receiver buffer to be loaded. A BRKDT interrupt can occur even if the receiver SLEEP bit is set to 1. BRKDT is cleared by an active SW RESET, SCIRST bit, or by a system reset. It is not cleared by receipt of a character after the break is detected. If Break Detect (BRKDT) is set, then RXRDY won't be set and there will be no further interrupts after the first interrupt where there is an error detected if a SW reset, channel reset, or system reset is not performed. In order to receive more characters, the SCI must be reset by toggling the SW RESET bit, channel reset (SCIRST), or by a system reset. NOTE: If your system is susceptible to break detects, ensure that you have a pull-up resistor on the SCI-RX pin to provide proper return-to-high signal behavior and noise immunity. NOTE: To monitor a break detect, place an oscilloscope on the C2000 SCI-RX line and monitor for a low-signal greater than 9.625 bits wide. If this is found and a break is not expected, please correct the software in the other device that is transmitting to this C2000 device. There should never be a low-signal greater than 9.625 bits wide on the SCI-RX line of the C2000 device unless a break detect is being transmitted purposely. Reset type: SYSRSn 0h (R/W) = No break condition 1h (R/W) = Break condition occurred
4	FE	R	0h	SCI framing-error flag. The SCI sets this bit when an expected stop bit is not found. Only the first stop bit is checked. The missing stop bit indicates that synchronization with the start bit has been lost and that the character is incorrectly framed. The FE bit is reset by a clearing of the SW RESET bit, channel reset (SCIRST), or by a system reset. NOTE: FE will be flagged prior to BRKDT, except when RX is in sleep mode. In sleep mode, when there is no RX WAKEUP and RXD line is low for greater than 10 bits, BRKDT will be flagged while FE will not be flagged. Reset type: SYSRSn 0h (R/W) = No framing error detected 1h (R/W) = Framing error detected

Table 21-13. SCIRXST Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	OE	R	0h	SCI overrun-error flag. The SCI sets this bit when a character is transferred into registers SCIRXEMU and SCIRXBUF before the previous character is fully read by the CPU or DMAC. The previous character is overwritten and lost. The OE flag bit is reset by an active SW RESET, channel reset (SCIRST), or a system reset. Reset type: SYSRSn 0h (R/W) = No overrun error detected 1h (R/W) = Overrun error detected
2	PE	R	0h	SCI parity-error flag. This flag bit is set when a character is received with a mismatch between the number of 1s and its parity bit. The address bit is included in the calculation. If parity generation and detection is not enabled, the PE flag is disabled and read as 0. The PE bit is reset by an active SW RESET, channel reset (SCIRST), or a system reset. Reset type: SYSRSn 0h (R/W) = No parity error or parity is disabled 1h (R/W) = Parity error is detected
1	RXWAKE	R	0h	Receiver wake-up-detect flag Reset type: SYSRSn 0h (R/W) = No detection of a receiver wake-up condition 1h (R/W) = A value of 1 in this bit indicates detection of a receiver wake-up condition. In the address-bit multiprocessor mode (SCICCR.3 = 1), RXWAKE reflects the value of the address bit for the character contained in SCIRXBUF. In the idle-line multiprocessor mode, RXWAKE is set if the SCIRXD data line is detected as idle. RXWAKE is a read-only flag, cleared by one of the following: - The transfer of the first byte after the address byte to SCIRXBUF (only in non-FIFO mode) - The reading of SCIRXBUF - An active SW RESET - Channel reset (SCIRST) - A system reset
0	RESERVED	R	0h	Reserved

21.15.2.7 SCIRXEMU Register (Offset = 6h) [Reset = 0000h]

SCIRXEMU is shown in Figure 21-17 and described in Table 21-14.

Return to the Summary Table.

Normal SCI data-receive operations read the data received from the SCIRXBUF register. The SCIRXEMU register is used principally by the emulator (EMU) because it can continuously read the data received for screen updates without clearing the RXRDY flag. SCIRXEMU is cleared by a system reset. This is the register that should be used in an emulator watch window to view the contents of the SCIRXBUF register. SCIRXEMU is not physically implemented
it is just a different address location to access the SCIRXBUF register without clearing the RXRDY flag.

Figure 21-17. SCIRXEMU Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
ERXDT							
R-0h							

Table 21-14. SCIRXEMU Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	ERXDT	R	0h	Receive emulation buffer data Reset type: SYSRSn

21.15.2.8 SCIRXBUF Register (Offset = 7h) [Reset = 0000h]

SCIRXBUF is shown in Figure 21-18 and described in Table 21-15.

Return to the Summary Table.

When the current data received is shifted from RXSHF to the receiver buffer, flag bit RXRDY is set and the data is ready to be read. If the RXBKINTENA bit (SCICTL2.1) is set, this shift also causes an interrupt. When SCIRXBUF is read, the RXRDY flag is reset. SCIRXBUF is cleared by a system reset.

Figure 21-18. SCIRXBUF Register

15	14	13	12	11	10	9	8
SCIFFFE	SCIFFPE	RESERVED					
R-0h	R-0h	R-0h					
7	6	5	4	3	2	1	0
SAR							
R-0h							

Table 21-15. SCIRXBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SCIFFFE	R	0h	SCIFFFE. SCI FIFO Framing error flag bit (applicable only if the FIFO is enabled) Note: 'SCIFFFE' is meant to serve as a flag for the specific set of data being received/read in the SCIRXBUF register. Each set of data received into the FIFO will have this information. The 'FE' bit within the SCIRXST register can be thought off as high level error flag where the flag will get set if any data that has been received has a framing error. Reset type: SYSRSn 0h (R/W) = No frame error occurred while receiving the character, in bits 7-0. This bit is associated with the character on the top of the FIFO. 1h (R/W) = A frame error occurred while receiving the character in bits 7-0. This bit is associated with the character on the top of the FIFO.
14	SCIFFPE	R	0h	SCIFFPE. SCI FIFO parity error flag bit (applicable only if the FIFO is enabled) Note: 'SCIFFPE' is meant to serve as a flag for the specific set of data being received/read in the SCIRXBUF register. Each set of data received into the FIFO will have this information. The 'PE' bit within the SCIRXST register can be thought off as high level error flag where the flag will get set if any data that has been received has a parity error. Note: If the parity is changed in the middle of data reception, the SCI module will not reinterpret the data with the new parity or other settings that may have changed. Therefore, changing the parameter, the FIFO should be cleared or the user should acknowledge that there will most likely be errors in the data caused by the change. Note: If RX parity errors are occurring intermittently this could be due to the length of the SCI ISR. To help prevent this, ensure that interrupt nesting is limited, increase the SCI interrupt priority, and move as much of the processing as possible out of the ISR (to reduce ISR time to the absolute minimum). Reset type: SYSRSn 0h (R/W) = No parity error occurred while receiving the character, in bits 7-0. This bit is associated with the character on the top of the FIFO. 1h (R/W) = A parity error occurred while receiving the character in bits 7-0. This bit is associated with the character on the top of the FIFO.
13-8	RESERVED	R	0h	Reserved

Table 21-15. SCIRXBUF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	SAR	R	0h	Receive Character bits Reset type: SYSRSn

21.15.2.9 SCITXBUF Register (Offset = 9h) [Reset = 0000h]

SCITXBUF is shown in Figure 21-19 and described in Table 21-16.

Return to the Summary Table.

Data bits to be transmitted are written to SCITXBUF. These bits must be rightjustified because the leftmost bits are ignored for characters less than eight bits long. The transfer of data from this register to the TXSHF transmitter shift register sets the TXRDY flag (SCICTL2.7), indicating that SCITXBUF is ready to receive another set of data. If bit TXINTENA (SCICTL2.0) is set, this data transfer also causes an interrupt.

Figure 21-19. SCITXBUF Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TXDT							
R/W-0h							

Table 21-16. SCITXBUF Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	TXDT	R/W	0h	Transmit data buffer Reset type: SYSRSn

21.15.2.10 SCIFFTX Register (Offset = Ah) [Reset = A000h]

SCIFFTX is shown in Figure 21-20 and described in Table 21-17.

Return to the Summary Table.

SCIFFTX controls the transmit FIFO interrupt, FIFO enhancements, and reset for the SCI transmit and receive channels.

Figure 21-20. SCIFFTX Register

15	14	13	12	11	10	9	8
SCIRST	SCIFFENA	TXFIFORESET	TXFFST				
R/W-1h	R/W-0h	R/W-1h	R-0h				
7	6	5	4	3	2	1	0
TXFFINT	TXFFINTCLR	TXFFIENA	TXFFIL				
R-0h	R-0/W1S-0h	R/W-0h	R/W-0h				

Table 21-17. SCIFFTX Register Field Descriptions

Bit	Field	Type	Reset	Description
15	SCIRST	R/W	1h	SCI Reset 0 A write of 0 will cause a SW RESET + a RESET of TXFFINT and RXFFINT, essentially clearing TX/RX FIFO content. The SCI will be held in reset until a write of 1. Additionally it resets the RXFFOVF, PE, OE, FE, RXERROR, BRKDET, RXRDY, and RXWAKE flags. It will also set TXRDY and TXEMPTY bits as 1. 1 SCI FIFO can resume transmit or receive. SCIRST should be 1 even for Autobaud logic to work. Reset type: SYSRSn
14	SCIFFENA	R/W	0h	SCI FIFO enable Reset type: SYSRSn 0h (R/W) = SCI FIFO enhancements are disabled 1h (R/W) = SCI FIFO enhancements are enabled
13	TXFIFORESET	R/W	1h	Transmit FIFO reset Reset type: SYSRSn 0h (R/W) = Reset the FIFO pointer to zero and hold in reset 1h (R/W) = Re-enable transmit FIFO operation
12-8	TXFFST	R	0h	FIFO status Reset type: SYSRSn 0h (R/W) = Transmit FIFO is empty 1h (R/W) = Transmit FIFO has 1 words 2h (R/W) = Transmit FIFO has 2 words 3h (R/W) = Transmit FIFO has 3 words 4h (R/W) = Transmit FIFO has 4 words 5h (R/W) = Transmit FIFO has 5 words 6h (R/W) = Transmit FIFO has 6 words 7h (R/W) = Transmit FIFO has 7 words 8h (R/W) = Transmit FIFO has 8 words 9h (R/W) = Transmit FIFO has 9 words Ah (R/W) = Transmit FIFO has 10 words Bh (R/W) = Transmit FIFO has 11 words Ch (R/W) = Transmit FIFO has 12 words Dh (R/W) = Transmit FIFO has 13 words Eh (R/W) = Transmit FIFO has 14 words Fh (R/W) = Transmit FIFO has 15 words 10h (R/W) = Transmit FIFO has 16 words
7	TXFFINT	R	0h	Transmit FIFO interrupt Reset type: SYSRSn 0h (R/W) = TXFIFO interrupt has not occurred, read-only bit 1h (R/W) = TXFIFO interrupt has occurred, read-only bit

Table 21-17. SCIFFTX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	TXFFINTCLR	R-0/W1S	0h	Transmit FIFO clear Reset type: SYSRSn 0h (R/W) = Write 0 has no effect on TXFIFINT flag bit, Bit reads back a zero 1h (R/W) = Write 1 to clear TXFFINT flag in bit 7
5	TXFFIENA	R/W	0h	Transmit FIFO interrrupt enable Reset type: SYSRSn 0h (R/W) = TX FIFO interrupt is disabled 1h (R/W) = TX FIFO interrupt is enabled. This interrupt is triggered whenever the transmit FIFO status (TXFFST) bits match (equal to or less than) the interrupt trigger level bits TXFFIL (bits 4-0).
4-0	TXFFIL	R/W	0h	TXFFIL4-0 Transmit FIFO interrupt level bits. The transmit FIFO generates an interrupt whenever the FIFO status bits (TXFFST4-0) are less than or equal to the FIFO level bits (TXFFIL4-0). The maximum value that can be assigned to these bits to generate an interrupt cannot be more than the depth of the TX FIFO. The default value of these bits after reset is 00000b. Users should set TXFFIL to best fit their application needs by weighing between the CPU overhead to service the ISR and the best possible usage of SCI bus bandwidth. Reset type: SYSRSn

21.15.2.11 SCIFFRX Register (Offset = Bh) [Reset = 201Fh]

SCIFFRX is shown in Figure 21-21 and described in Table 21-18.

Return to the Summary Table.

SCIFFRX controls the receive FIFO interrupt, receive FIFO reset, and status of the receive FIFO overflow.

Figure 21-21. SCIFFRX Register

15	14	13	12	11	10	9	8
RXFFOVF	RXFFOVRCLR	RXFIFORESET	RXFFST				
R-0h	R-0/W1S-0h	R/W-1h	R-0h				
7	6	5	4	3	2	1	0
RXFFINT	RXFFINTCLR	RXFFIENA	RXFFIL				
R-0h	W-0h	R/W-0h	R/W-1Fh				

Table 21-18. SCIFFRX Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RXFFOVF	R	0h	Receive FIFO overflow. This will function as flag, but cannot generate interrupt by itself. This condition will occur while receive interrupt is active. Receive interrupts should service this flag condition. This bit is cleared by RXFFOVRCLR, a channel reset (SCIRST), or a system reset. Reset type: SYSRSn 0h (R/W) = Receive FIFO has not overflowed, read-only bit 1h (R/W) = Receive FIFO has overflowed, read-only bit. More than 16 words have been received in to the FIFO, and the first received word is lost
14	RXFFOVRCLR	R-0/W1S	0h	RXFFOVF clear Note: Both RXFFIL and RXFFOVF flags are ORed together, so they need to be cleared at the same time (RXFFINTCLR & RXFFOVRCLR) during overflow scenarios else it will prevent further interrupts from occurring. Reset type: SYSRSn 0h (R/W) = Write 0 has no effect on RXFFOVF flag bit, Bit reads back a zero 1h (R/W) = Write 1 to clear RXFFOVF flag in bit 15
13	RXFIFORESET	R/W	1h	Receive FIFO reset Reset type: SYSRSn 0h (R/W) = Write 0 to reset the FIFO pointer to zero, and hold in reset. 1h (R/W) = Re-enable receive FIFO operation

Table 21-18. SCIFFRX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-8	RXFFST	R	0h	FIFO status Reset type: SYSRSn 0h (R/W) = Receive FIFO is empty 1h (R/W) = Receive FIFO has 1 words 2h (R/W) = Receive FIFO has 2 words 3h (R/W) = Receive FIFO has 3 words 4h (R/W) = Receive FIFO has 4 words 5h (R/W) = Receive FIFO has 5 words 6h (R/W) = Receive FIFO has 6 words 7h (R/W) = Receive FIFO has 7 words 8h (R/W) = Receive FIFO has 8 words 9h (R/W) = Receive FIFO has 9 words Ah (R/W) = Receive FIFO has 10 words Bh (R/W) = Receive FIFO has 11 words Ch (R/W) = Receive FIFO has 12 words Dh (R/W) = Receive FIFO has 13 words Eh (R/W) = Receive FIFO has 14 words Fh (R/W) = Receive FIFO has 15 words 10h (R/W) = Receive FIFO has 16 words
7	RXFFINT	R	0h	Receive FIFO interrupt Reset type: SYSRSn 0h (R/W) = RXFIFO interrupt has not occurred, read-only bit 1h (R/W) = RXFIFO interrupt has occurred, read-only bit
6	RXFFINTCLR	W	0h	Receive FIFO interrupt clear Note: Both RXFFIL and RXFFOVF flags are ORed together, so they need to be cleared at the same time (RXFFINTCLR & RXFFOVRCLR) during overflow scenarios else it will prevent further interrupts from occurring. Reset type: SYSRSn 0h (R/W) = Write 0 has no effect on RXFIFINT flag bit. Bit reads back a zero. 1h (R/W) = Write 1 to clear RXFFINT flag in bit 7
5	RXFFIENA	R/W	0h	Receive FIFO interrupt enable Reset type: SYSRSn 0h (R/W) = RX FIFO interrupt is disabled 1h (R/W) = RX FIFO interrupt is enabled. This interrupt is triggered whenever the receive FIFO status (RXFFST) bits match (equal to or greater than) the interrupt trigger level bits RXFFIL (bits 4-0).
4-0	RXFFIL	R/W	1Fh	Receive FIFO interrupt level bits The receive FIFO generates an interrupt whenever the FIFO status bits (RXFFST4-0) are greater than or equal to the FIFO level bits (RXFFIL4-0). The maximum value that can be assigned to these bits to generate an interrupt cannot be more than the depth of the RX FIFO. The default value of these bits after reset is 11111b. Users should set RXFFIL to best fit their application needs by weighing between the CPU overhead to service the ISR and the best possible usage of received SCI data. Reset type: SYSRSn

21.15.2.12 SCIFFCT Register (Offset = Ch) [Reset = 0000h]

SCIFFCT is shown in Figure 21-22 and described in Table 21-19.

Return to the Summary Table.

SCIFFCT contains the status of auto-baud detect, clears the auto-baud flag, and calibrate for A-detect bit.

Figure 21-22. SCIFFCT Register

15	14	13	12	11	10	9	8
ABD	ABDCLR	CDC	RESERVED				
R-0h	W-0h	R/W-0h	R-0h				
7	6	5	4	3	2	1	0
FFTXDLY							
R/W-0h							

Table 21-19. SCIFFCT Register Field Descriptions

Bit	Field	Type	Reset	Description
15	ABD	R	0h	Auto-baud detect (ABD) bit Reset type: SYSRSn 0h (R/W) = Auto-baud detection is not complete. 'A','a' character has not been received successfully. 1h (R/W) = Auto-baud hardware has detected 'A' or 'a' character on the SCI receive register. Auto-detect is complete.
14	ABDCLR	W	0h	ABD-clear bit Reset type: SYSRSn 0h (R/W) = Write 0 has no effect on ABD flag bit. Bit reads back a zero. 1h (R/W) = Write 1 to clear ABD flag in bit 15.
13	CDC	R/W	0h	CDC calibrate A-detect bit Reset type: SYSRSn 0h (R/W) = Disables auto-baud alignment 1h (R/W) = Enables auto-baud alignment
12-8	RESERVED	R	0h	Reserved
7-0	FFTXDLY	R/W	0h	FIFO transfer delay. These bits define the delay between every transfer from FIFO transmit bufferto transmit shift register. The delay is defined in the number of SCI serial baud clock cycles. The 8 bit register could define a minimum delay of 0 baud clock cycles and a maximum of 256 baud clock cycles In FIFO mode, the buffer (TXBUF) between the shift register and the FIFO should be filled only after the shift register has completed shifting of the last bit. This is required to pass on the delay between transfers to the data stream. In FIFO mode, TXBUF should not be treated as one additional level of buffer. The delayed transmit feature will help to create an auto-flow scheme without RTS/CTS controls as in standard UARTS. When SCI is configured for one stop-bit, delay introduced by FFTXDLY between one frame and the next frame is equal to number of baud clock cycles that FFTXDLY is set to. When SCI is configured for two stop-bits, delay introduced by FFTXDLY between one frame and the next frame is equal to number of baud clock cycles that FFTXDLY is set to minus 1. Reset type: SYSRSn

21.15.2.13 SCIPRI Register (Offset = Fh) [Reset = 0000h]

SCIPRI is shown in Figure 21-23 and described in Table 21-20.

Return to the Summary Table.

SCIPRI determines what happens when an emulation suspend event occurs.

Figure 21-23. SCIPRI Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			FREESOFT		RESERVED		
R-0h			R/W-0h		R-0h		

Table 21-20. SCIPRI Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-5	RESERVED	R	0h	Reserved
4-3	FREESOFT	R/W	0h	These bits determine what occurs when an emulation suspend event occurs (for example, when the debugger hits a breakpoint). The peripheral can continue whatever it is doing (free-run mode), or if in stop mode, it can either stop immediately or stop when the current operation (the current receive/transmit sequence) is complete. Reset type: SYSRSn 0h (R/W) = Immediate stop on suspend 1h (R/W) = Complete current receive/transmit sequence before stopping 2h (R/W) = Free run 3h (R/W) = Free run
2-0	RESERVED	R	0h	Reserved

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from July 1, 2025 to October 31, 2025 (from Revision * (July 2025) to Revision A (October 2025))	Page
• *Reference Summary Table*: Replaced CONFIG8 bitfield name with ANAREFSEL.	824
• Changed Note in Section 12.2.1	855
• Changed Figure 12-3	860
• Changed Figure 12-9	871
• Changed Table 12-9	879
• Changed Table 12-10 and Table 12-11	879

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025