OMAP5910 Dual-Core Processor Clock Generation and System Reset Management Reference Guide

Literature Number: SPRU678A
October 2003 − Revised July 2005

Preface

Read This First

About This Manual

This document describes clock generation and system reset for the OMAP5910 multimedia processor.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.

Related Documentation From Texas Instruments

The following documents describe the OMAP5910 device and related peripherals. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

OMAP5910 Dual-Core Processor MPU Subsystem Reference Guide (literature number SPRU671)

OMAP5910 Dual-Core Processor DSP Subsystem Reference Guide (literature number SPRU672)

OMAP5910 Dual-Core Processor Memory Interface Traffic Controller Reference Guide (literature number SPRU673)

OMAP5910 Dual-Core Processor System DMA Controller Reference Guide (literature number SPRU674)

OMAP5910 Dual-Core Processor LCD Controller Reference Guide (literature number SPRU675)

OMAP5910 Dual-Core Processor Universal Asynchronous Receiver/Transmitter (UART) Devices Reference Guide (literature number SPRU676)

OMAP5910 Dual-Core Processor Universal Serial Bus (USB) and Frame Adjustment Counter (FAC) Reference Guide (literature number SPRU677)

OMAP5910 Dual-Core Processor Clock Generation and System Reset Management Reference Guide (literature number SPRU678)

OMAP5910 Dual-Core Processor General-Purpose Input/Output (GPIO) Reference Guide (literature number SPRU679)

OMAP5910 Dual-Core Processor MMC/SD Reference Guide (literature number SPRU680)

OMAP5910 Dual-Core Processor Inter-Integrated Circuit (I2C) Controller Reference Guide (literature number SPRU681)

OMAP5910 Dual-Core Processor Timer Reference Guide (literature number SPRU682)

OMAP5910 Dual-Core Processor Inter-Processor Communication Reference Guide (literature number SPRU683)

OMAP5910 Dual-Core Processor Camera Interface Reference Guide (literature number SPRU684)

OMAP5905 Dual-Core Processor Multichannel Serial Interface (MCSI) Reference Guide (literature number SPRU685)

OMAP5910 Dual-Core Processor Micro-Wire Interface Reference Guide (literature number SPRU686)

OMAP5910 Dual-Core Processor Real-Time Clock (RTC) Reference Guide (literature number SPRU687)

OMAP5910 Dual-Core Processor HDQ/1-Wire Interface Reference Guide (literature number SPRU688)

OMAP5910 Dual-Core Processor PWL, PWT, and LED Peripheral Reference Guide (literature number SPRU689)

OMAP5910 Dual-Core Processor Multichannel Buffered Serial Port (McBSP) Reference Guide (literature number SPRU708)

Trademarks

OMAP and the OMAP symbol are trademarks of Texas Instruments.

Contents

Figures

Tables

This page intentionally left blank.

Clock Generation and System Reset Management

This document describes clock generation and system reset for the OMAP5910 multimedia processor.

1 Introduction

The clock generator and reset management module supplies clocks and resets to the entire OMAP5910 device.

Figure 1 shows the OMAP5910 device with the clock generator and reset-management area highlighted. Figure 2 shows the OMAP5910 clock scheme.

1.1 Clock Generation and System Reset Control

In the OMAP5910 device, clock generation and system reset are controlled by several modules, as shown in Figure 3.

There are three major components of this circuitry: the ultralow-power device (ULPD), the reset-management module, and the clock-generation and management module.

Figure 1. OMAP5910 Device Clock and Reset Management

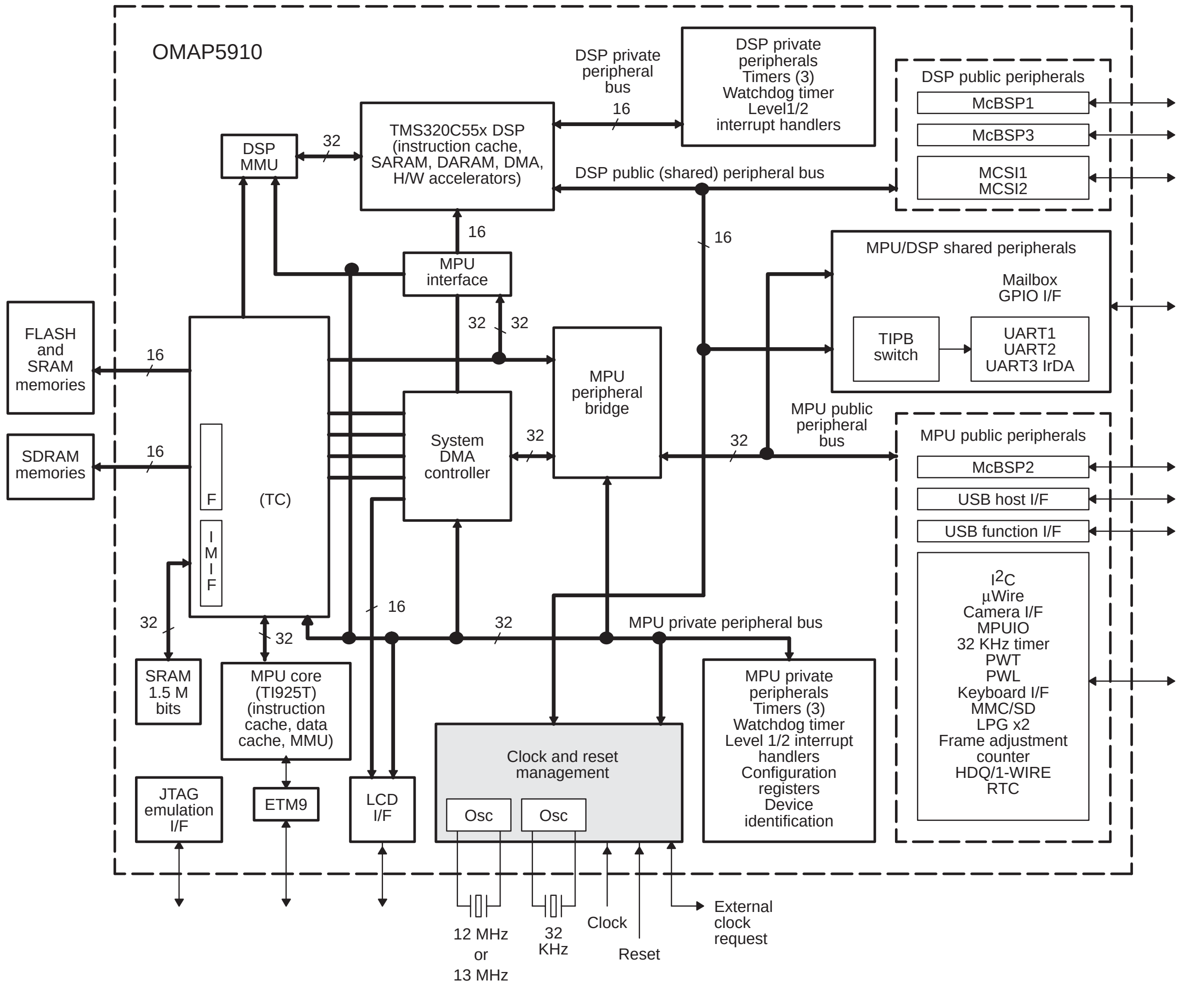

OMAP5910
DSP private peripheral bus
DSP private peripherals Timers (3) Watchdog timer Level1/2 interrupt handlers
DSP public peripherals
McBSP1
McBSP3
MCSI1 MCSI2
DSP MMU
TMS320C55x DSP (instruction cache, SARAM, DARAM, DMA, H/W accelerators)
DSP public (shared) peripheral bus
MPU interface
MPU/DSP shared peripherals
Mailbox GPIO I/F
TIPB switch
UART1 UART2 UART3 IrDA
FLASH and SRAM memories
SDRAM memories
F
(TC)
I M I F
System DMA controller
MPU peripheral bridge
MPU public peripheral bus
MPU public peripherals
McBSP2
USB host I/F
USB function I/F
I2C µWire Camera I/F MPUIO 32 KHz timer PWT PWL Keyboard I/F MMC/SD LPG x2 Frame adjustment counter HDQ/1-WIRE RTC
MPU private peripheral bus
SRAM 1.5 M bits
MPU core (TI925T) (instruction cache, data cache, MMU)
MPU private peripherals Timers (3) Watchdog timer Level 1/2 interrupt handlers Configuration registers Device identification
JTAG emulation I/F
ETM9
LCD I/F
Clock and reset management
Osc
Osc
12 MHz or 13 MHz
32 KHz
Clock
Reset
External clock request
16
32
16
16
32
32
16
16
32
32
32
16
32
32

Figure 2. *OMAP5910 Clock Scheme*

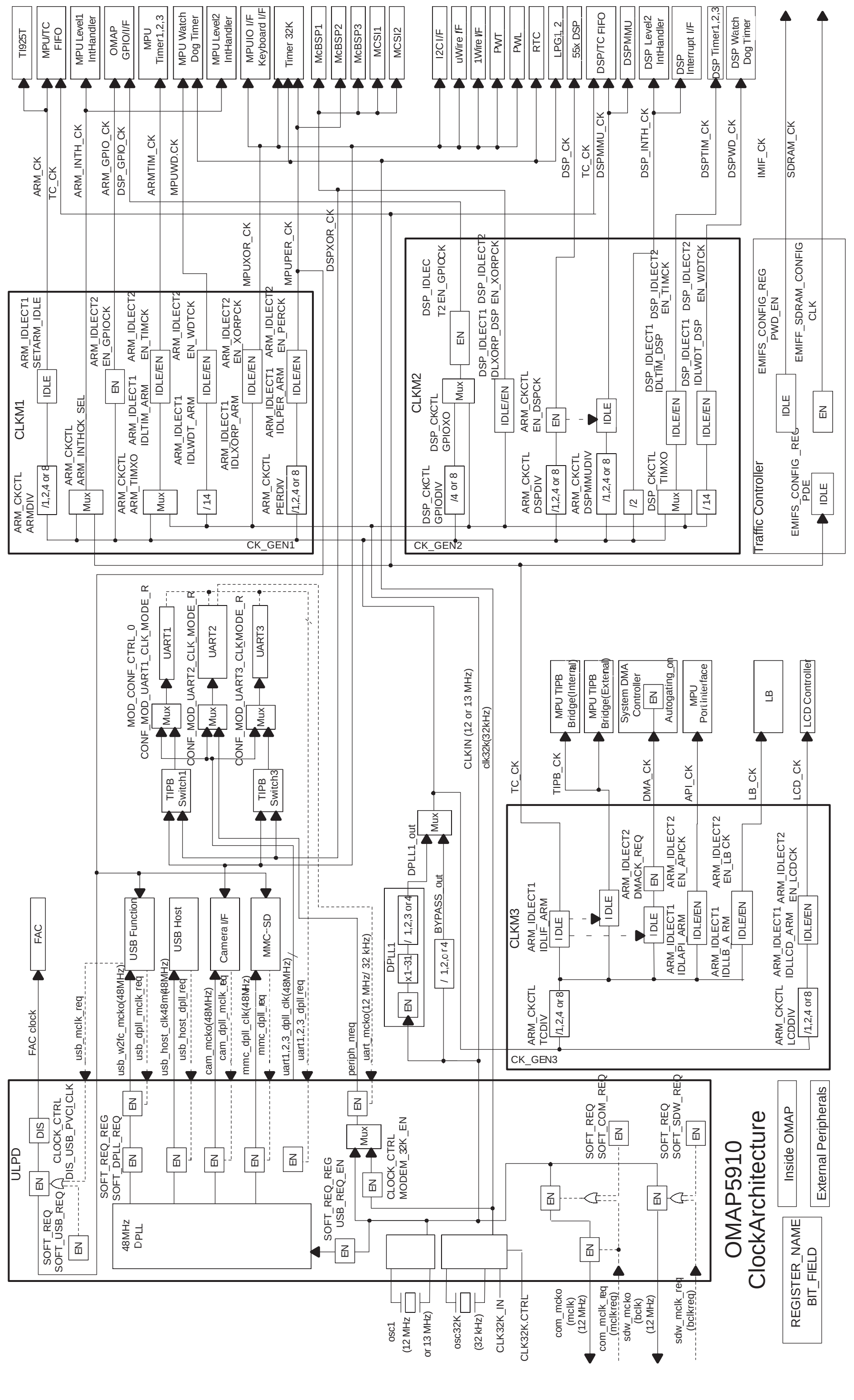
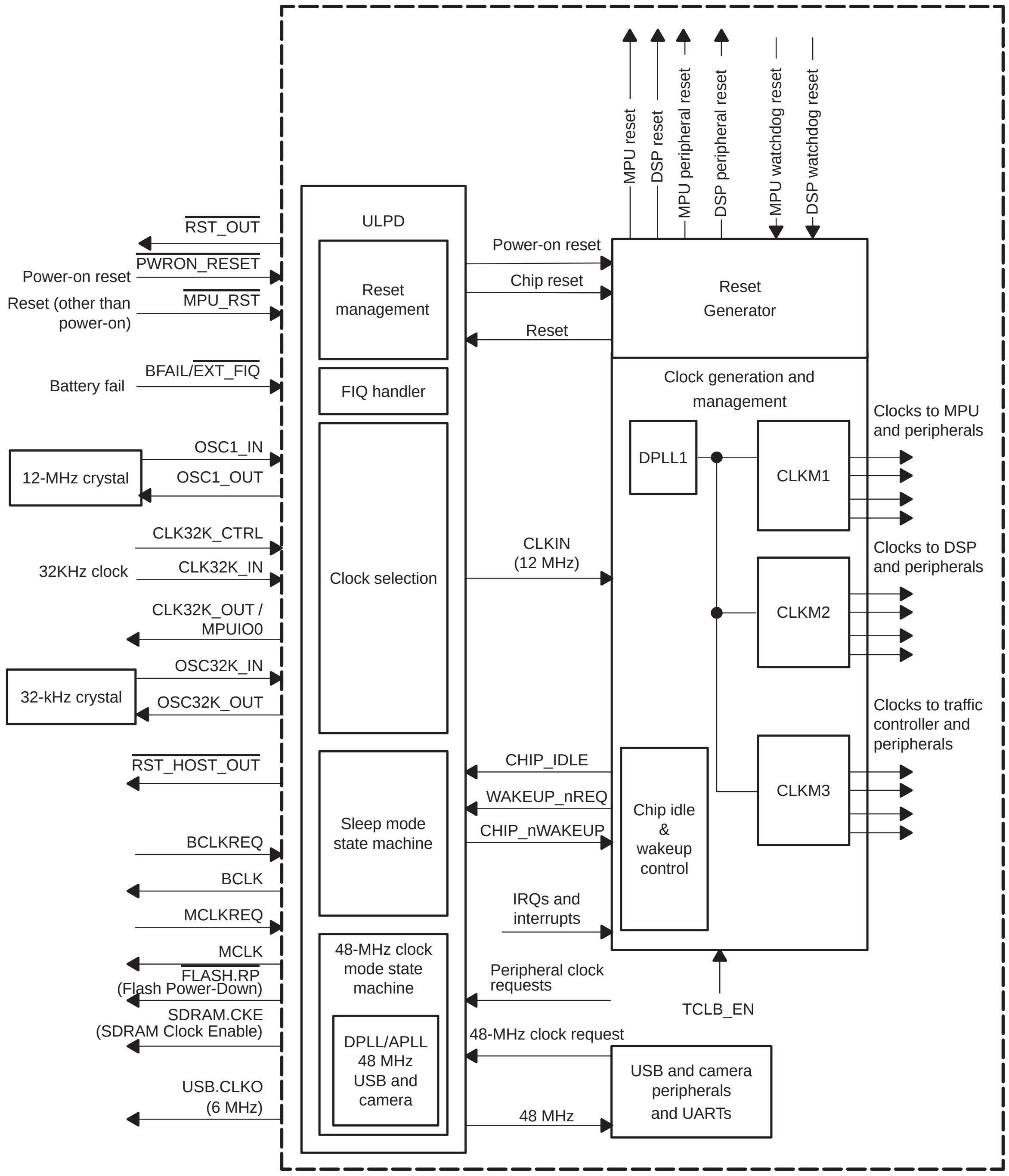


Figure 3. Modules Controlling Clock and Reset Management



1.1.1 ULPD Module

The ULPD module is an embedded peripheral controlled by the internal MPU with the following functions:

- ❑ Performs the state transition of the different power modes:
  - ■ Awake: The 32-kHz and 12-MHz clocks are on, and 12 MHz is fed into the clock-generation module (those PLLs can be turned on or off).
  - ■ Big-sleep mode: The 32-kHz and 12-MHz clocks are on, 12 MHz is not fed into the clock-generation module, and the PLLs are off.
  - ■ Deep-sleep mode: The 32-kHz clock is on, and the 12-MHz clock and the PLLs are off.
- ❑ Performs the power-on reset of the chip
- ❑ Calibrates an external quartz-based oscillator (32 kHz)
- ❑ Performs the wake up of the 12-MHz OSC1 oscillator to provide the OSC1 clock to an external device. An external-clock request or peripheral-wake-up request turns on the OSC1 oscillator but neither request is not treated as an interrupt.
- ❑ Performs a 12-MHz/32-kHz switch for the peripherals that need to switch to 32 kHz
- ❑ Generates the functional-reset signal used by the reset module
- ❑ Manages the 48-MHz DPLL and the APLL on/off
- ❑ Processes the battery-failed signal to generate the external-shutdown signal ($\overline{\text{RST\_HOST\_OUT}}$)

1.1.2 Reset Module

The reset module has the following functions:

- ❑ Provides the internal-global reset and the software reset
- ❑ Performs the reset control for the peripheral-bus peripherals
- ❑ Monitors the internal and the external reset (for example, the watchdog-timer time-out)
- ❑ Monitors the system and reset status

1.1.3 Clock-Generation and Management Module

The clock-generation and management module provides the following features:

- Programmable-clocking scheme (synchronous and synchronous scalable modes) and power-up defaults to the fully-synchronous mode
- Setup and configuration can be controlled by both the DSP and the MPU processors
- A single reference-clock input to one DPLL from an external source (CLKIN). The PLL modes are configurable as follows: lock, bypass, and idle.
- Programmable clock, from the CLKM1 (see Section 1.1.5, *Clock Domains*), with the clock and idle-control capability to the MPU and its subsystem:
  - GPIO
  - Timers
  - Other peripherals
- Programmable clock, from the CLKM2 (see Section 1.1.5, *Clock Domains*), with the clock and idle-control capability to the DSP and its subsystem:
  - GPIO
  - Timers
  - Other peripherals
- Programmable clock, from the CLKM3 (see Section 1.1.5, *Clock Domains*), with the clock and idle-control capability to the memory-interface traffic controller (TC), including the following modules:
  - MPU interface (MPUI)
  - System DMA controller
  - LCD controller
  - Local bus
  - MPU peripheral bridges
  - Two internal MPU-TI-peripheral bridges to minimize access latency
- Programmable power-saving and idle-mode controls for the MPU, the DSP, the TC, and their respective subdomains
- Low-frequency clocks (reference clock/14) to supply the watchdog timers for the DSP and MPU
- DMA clock-request mechanism (provides the DMA clock during data transfer only)

- Power control for an external device reset/power on (flash)
- Idle-sequence controls (MPU-clock domain, DSP-clock domain, and TC-clock domain)
- Programmable idle modes (MPU and DSP) for different applications
- Wake up is initiated by interrupts (MPU, DSP, and TCLB_EN pin) or DMA requests (TC and peripheral bus) in the idle mode
- Unmasked-interrupt events are enabled to wake up the device during idle modes

1.1.4 Memory-Mapped Registers

The application program controls the clock generation, reset, and power-saving modes via a set of memory-mapped registers (nine 16-bit registers for the MPU subsystem and seven 16-bit registers for the DSP subsystem). These registers are accessible by the MPU or the DSP processors.

The MPU is the master of the OMAP5910 device at all times, and it controls the activities in the MPU, the DSP, and the TC domains.

The DSP controls the DSP’s peripheral activities.

1.1.5 Clock Domains

The OMAP5910 device is partitioned into three clock domains, each with its own clock manager:

- MPU domain (CLKM1)
- DSP domain (CLKM2)
- TC domain (CLKM3)

The three clock domains use a common DPLL to provide a synchronous clock. The different clocking configurations are discussed in detail later in this document.

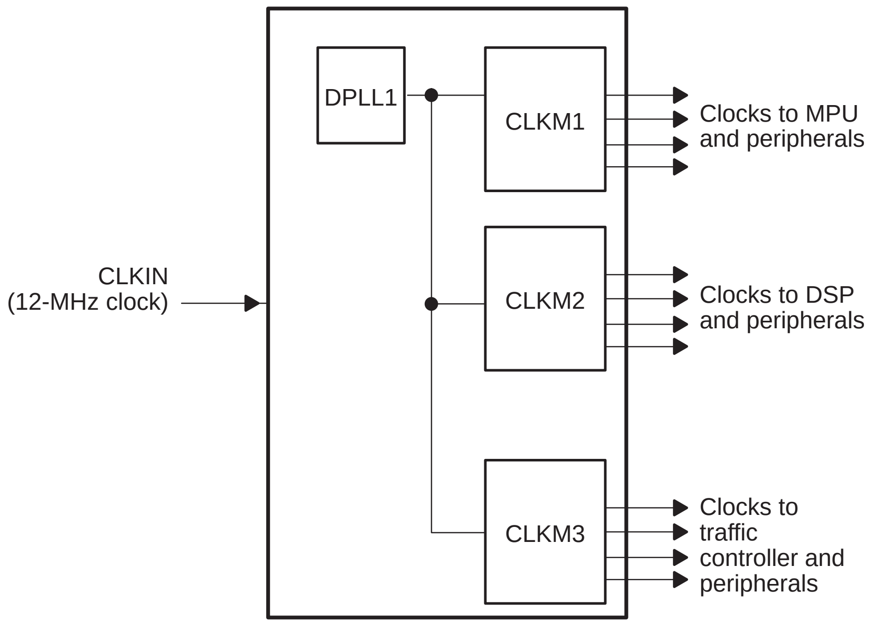
The external clock-source (OSC1_IN) frequency must be 12 MHz to ensure proper operation of the USB.

2 Clock Generation

Figure 4 shows the basic building blocks of the clock-generator and system reset module. This module consists of:

- ❑ One DPLL—frequency synthesizer (frequency locked but not phase locked)
- ❑ Control register file (CLKREG)—clock generator, system reset, idle, and wake-up controls
- ❑ Three CLKMs—clock generation and wake-up controls

Figure 4. Clock Generation and System Reset Module


CLKIN
(12-MHz clock)
DPLL1
CLKM1
CLKM2
CLKM3
Clocks to MPU and peripherals
Clocks to DSP and peripherals
Clocks to traffic controller and peripherals

2.1 Clocking Schemes

The clock generator supports two clocking schemes to provide performance flexibility and power-saving capabilities to the system. The clocking schemes are programmable. The power-up mode defaults to the fully-synchronous mode. Table 1 shows the clocking-scheme selection, and Table 2 shows the CLKM-source selection.

Table 1. Clocking Scheme Selection

Clock_Select	Clocking Scheme	Remarks
000	Full synchronous	Default, bypass the FIFO logic. TC=DSPMMU=MPU, DSP = 1x or 2x of the DSPMMU
001	Reserved	Do not use this setting
010	Synchronous scalable	Use the FIFO logic between the MPU and the TC, the DSP MMU and the TC
Others	Reserved	Do not use these settings

Note: In all of the above cases, the frequency of the DSP can be 1x or 2x that of the DSP MMU.

Table 2. CLKM−Source Selection—Set via the MPU−System−Status Register

Clock Select	Operating Mode	CLKM1 Input Clock Source	CLKM2 Input Clock Source	CLKM3 Input Clock Source	Remarks
000	Fully synchronous	DPLL1/N	DPLL1/O	DPLL1/N	Notes 1, 2, 4
010	Synchronous scalable	DPLL1/M	DPLL1/N	DPLL1/O	Notes 3, 4

Notes:
1) If the fully−synchronous mode is selected, the divide-down bits must be programmed so that the MPUDIV, the DSPMMUDIV and the TCDIV are all equal. Further, the DSPMMUDIV must be 1x or 1/2x that of the DSPDIV*.*
2) CLKGEN1 = CLKGEN3 = DPLL1/N, CLKGEN2 = CLKGEN1 or 2*CLKGEN1 = DPLL1/O
3) M, N =< O, and O is a multiple (1, 2, 4, 8) of M, N.
4) The DSP MMU cannot run above the maximum speed of the TC.

2.2 Operating Modes

The OMAP5910 device supports the following operating modes:

❑ Fully synchronous

The MPU, DSP MMU, and TC run at the same clock period, and the DSP MMU is 1x or 1/2x of the DSP. For example, the DPLL1 output can be 120 MHz; the MPU, DSP MMU, and TC can be 60 MHz; and the DSP can be 120 MHz.

On power up, the OMAP5910 device is always in the fully−synchronous mode, where the MPU, DSP MMU, TC, and DSP are all at the same speed.

❑ Synchronous scalable

The MPU, DSP MMU, and TC are synchronous, but the MPU and DSP MMU are multiples (1x, 2x, 4x, or 8x) of the TC. The DSP must be 1x or 2x of the DSP MMU. For example, the DPLL1 clock can be 120 MHz, the MPU can run at 120 MHz, the DSP MMU can run at 60 MHz, the TC can run at 30 MHz, and the DSP can run at 60 MHz or 120 MHz. In this mode, the clock-feeding mechanism (to each respective domain) is similar to that of the fully-synchronous mode, with the exception that the clocks are synchronous but are multiples of each other. The input clock is from the DPLL1, and the clock is multiplied/divided by the CLKM (1, 2, 3) as in the following example (assuming that the output of the DPLL1 is 120 MHz):

- CLKM1 output: 120 MHz/2
- CLKM2 output: 120 MHz/1
- CLKM3 output: 120 MHz/4

Divider circuitry is implemented in each CLKM.

Note:

In synchronous-scalable mode, the traffic controller clock must have the same or a slower frequency as the MPU and the DSPMMU clocks.

At reset, the fully-synchronous mode is selected (default). After the OMAP5910 device is up and running, the application software can write to the control registers via the CLOCK_SELECT[2:0] bits in the MPU-system-status register (ARM_SYSST) to switch to a desired mode of operation. However, system software should be used to save the context before switching modes. For information about the switching procedure, see Section 5, *Switching Clock Modes*.

The DSP_MMU clock must obey all of the following rules:

1) The TC-clock frequency always must be equal to or less than the DSP, DSPMMU, and MPU clocks.

2) The DSP_MMU clock must be 1x or 1/2x of the DSP clocks.

3) The DSPMMU_DIV can be /1, /2, /4, /8, but the TC_DIV and DSP_DIV must obey rules 1) and 2) immediately above.

4) The DSP_MMU-clock frequency cannot be more than the maximum speed of the TC.

2.3 External-Master Mode

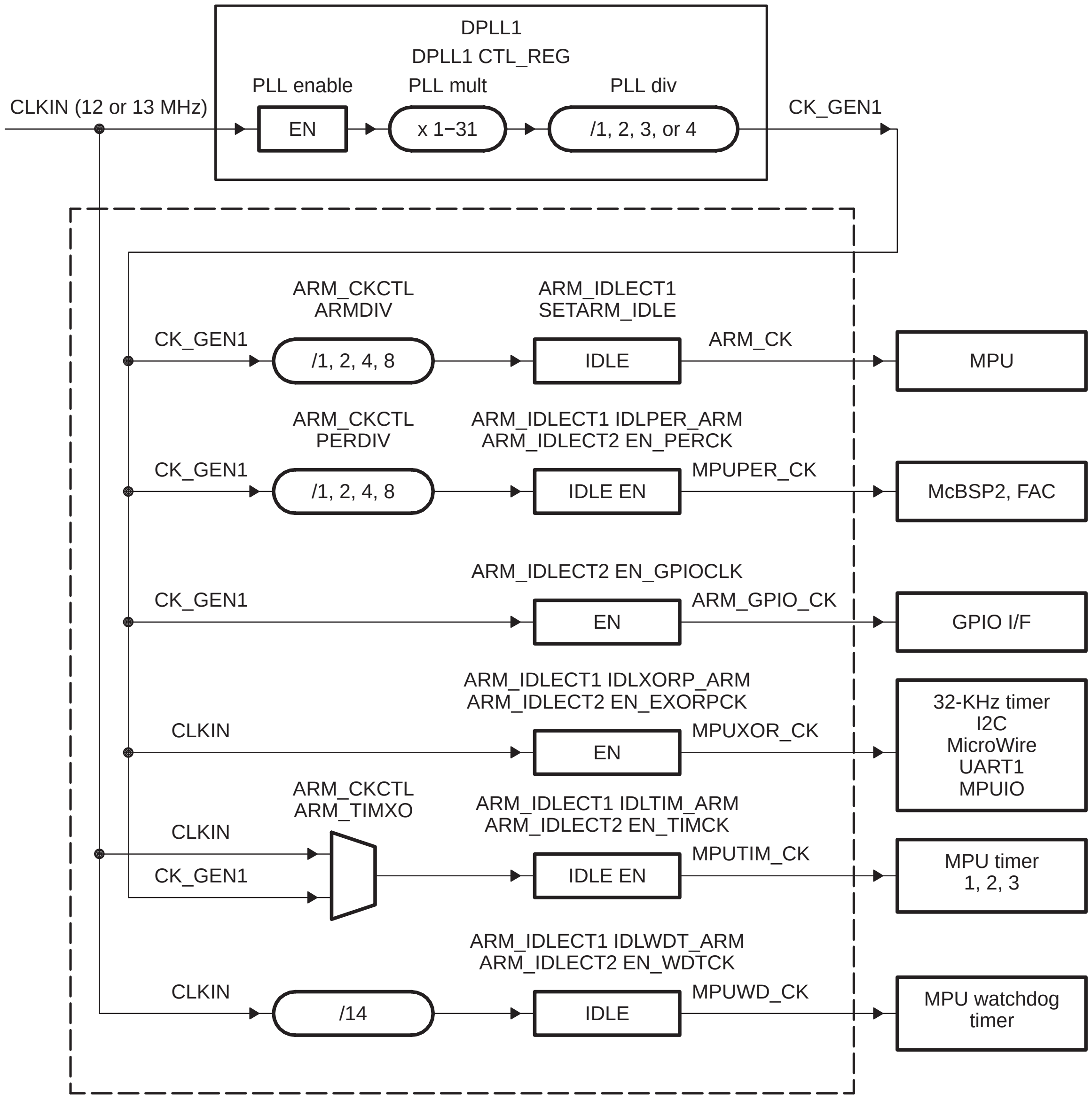
This mode allows a bypass of the 12-Mhz on-chip oscillator in systems where the 12-MHz clock is provided externally by a master device. The procedure for utilizing this mode is as follows:

1) During the power-on reset, the OMAP5910 device is in the deep-sleep mode:
   - The 12-MHz on-chip oscillator is disabled.
   - The MCLKREQ pin is an input.
2) After the power-on reset, the OMAP5910 device is awake.
   - The 12-MHz oscillator is bypassed.
   - The MCLKREQ pin is an input.
   - The 12-Mhz clock is provided externally.
3) Switch to external-master mode by setting FUNC_MUX_CTRL_B[20:18] = 001.
   - The 12-MHz oscillator is bypassed (disabled).
   - The MCLKREQ pin is now the EXT_MASTER_REQ pin, which drives to 1.
   - The 12-Mhz clock is provided externally.
4) If the OMAP5910 device switches into the deep-sleep mode:
   - The EXT_MASTER_REQ drives to 0 to indicate that the external 12-MHz clock is not needed.
   - The 12-MHz clock can then be switched off externally.

2.4 CLKM1

The CLKM1 controls the clock distribution and idle modes of the MPU subsystem, plus the associated private and public peripherals (see Figure 5).

Figure 5. MPU Clock Distribution



The MPU clock (see Figure 5) has the following domains (CLKM1):

- The MPU-processor clock: ARM_CK, which is CLK_GEN1 divided by 1, 2, 4, or 8, as programmed via the ARMDIV bits of the MPU-clock-control register (ARM_CKCTL). The idle mode of the MPU is controlled by the SETARM_IDLE bit of the MPU-idle-mode-entry-1 register (ARM_ IDLECT1).
- The MPU-peripheral clocks are:
  - The MPUXOR_CK, which is derived from CLKIN
  - The MPUPER_CK, which is CLK_GEN1 divided by 1, 2, 4 or 8, as programmed via the PERDIV bits of the MPU-clock-control register (ARM_CKCTL)

    The MPUPER_CLK clock is enabled by the EN_PERCK bit of the MPU-idle-mode-entry-2 register (ARM_IDLECT2) and the MPUX-OR_CK clock by the EN_XORPCK bit.
- The MPU-watchdog-timer clock (low frequency, derived from CLKIN/14): called either CK_CLKIN14 or MPUWD_CK. This clock is enabled by the EN_WDTCK bit of the MPU-idle-mode-entry-2 register (ARM_IDLECT2). The idle mode is controlled by the IDLWDT_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1). The clock cannot be disabled or idled while in the watchdog mode.
- The MPU-internal-timers clock: MPUTIM_CK, which is derived from either the CK_GEN1 or the CLKIN, as selected by the ARM_TIMXO bit of the MPU-clock-control register (ARM_CKCTL). The idle mode of the MPU timers is controlled by the IDLTIM_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1) and is enabled by the EN_TIMCK bit of the MPU-idle-mode-entry-2 register (ARM_IDLECT2).
- The MPU GPIO clock, MPU_GPIO_CLK, which is equal to the CK_GEN1. This clock is enabled by the EN_GPIOCK bit of the MPU-idle-mode-entry-2 register (ARM_IDLECT2).

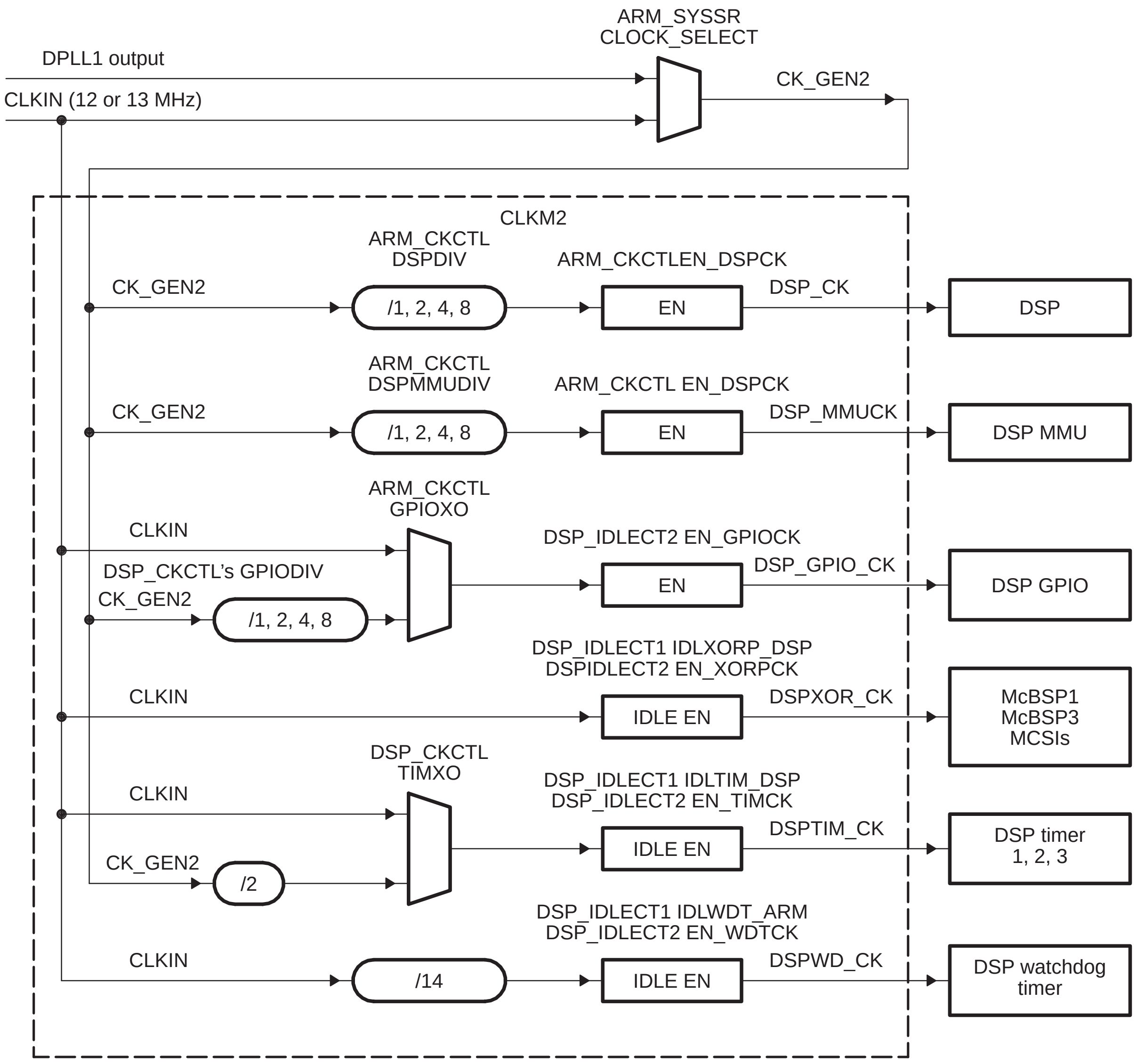
2.5 CLKM2

The CLKM2 controls the clock distribution and idle modes of the DSP subsystem plus the associated private and shared peripherals. As shown in Figure 6, the CLKM2 circuitry provides separate clock signals for the DSP internal peripherals (GPIO, watchdog timer, and timers) and the public peripherals.

Note:

The CK_GEN2 represents the output of the DPLL1 or the CLKIN, depending on the clocking mode enabled.

Figure 6. DSP Clock Distribution



The clock signals for each clock domain of the DSP subsystem are as follows:

- MPU-controlled:
  - The DSP-processor clock: DSP_CK, which is CK_GEN2 divided by 1, 2, 4, or 8, as programmed via the DSPDIV bits of the MPU clock control register (ARM_CKCTL). The enabling of the DSP_CK while the DSP is in the reset state is controlled by the EN_DSPCK bit of the MPU-clock-control register (ARM_CKCTL).
  - The DSP-MMU clock: DSPMMU_CK, as derived from the CK_GEN2 divided by 1, 2, 4, 8, as programmed by the DSPMMUDIV bits of the MPU-clock-control register (ARM_CKCTL). Take care in selecting clocking schemes so as not to exceed the maximum frequency of the DSP MMU. See the OMAP5910 device datasheet for absolute timing limits.
- DSP-controlled:
  - The DSP GPIO (DSP_GPIO_CK), as derived from the CK_GEN2 divided by 1, 2, 4, 8 (as programmed by the DSP_CKCTL GPIODIV) or CLKIN, as selected by the GPIOXO bit of the DSP-clock-control register (DSP_CKCTL). The clock is enabled by the EN_GPIOCLK bit of the DSP-idle-mode-entry-2 register (DSP_IDLECT2).
  - The DSP-public peripherals (McBSPs, MCSIs): DSPXOR_CK, which is derived from the CLKIN. The clock is enabled by the EN_XORPCK bit of the DSP-idle-mode-entry-2 register (DSP_IDLECT2). The idle mode is controlled by the IDLXORP_DSP bit of the DSP-idle-mode-entry-1 register (DSP_IDLECT1).
  - The DSP-internal timer: DSPTIM_CK, is selected from either the CK_GEN2/2 or the CLKIN via the DSP_TIMXO bit of the DSP-clock-control register (DSP_CKCTL). The clock is enabled by the EN_TIMCLK bit of the DSP-idle-mode-entry-2 register (DSP_IDLECT2). The idle mode is controlled by the IDLTIM_DSP bit of the DSP-idle-mode-entry-1 register (DSP_IDLECT1).
  - The DSP-watchdog timer (low frequency, derived from CLKIN divided by 14): DSPWD_CK. The clock is enabled by the EN_WDCLK bit of the DSP-idle-mode-entry-2 register (DSP_IDLECT2). The idle mode is controlled by the IDLWDT_DSP bit of the DSP-idle-mode-entry-1 register (DSP_IDLECT1). The watchdog-timer clock can only be disabled or idled when not in the watchdog mode.

After reset, the highest-frequency option (CK_GEN2 divided by 1) is selected for the GPIO and the DSPPER clocks. The software-application program can alter these divisors at any time during operation by writing to the GPIODIV or the DSP_PERDIV bits in the DSP-clock-control register (DSP_CKCTL).

The clock-generator output (CK_GEN2) delivers a 50%-duty cycle to the DSP subsystem clock-distribution module (CLKM2). This module provides additional clock scaling, routing, and idle/reset control to the DSP to individual components in the DSP clock domain.

The CK_GEN2 clock works in conjunction with the idle and the wake-up control logic to produce the DSP_CK-clock signal that drives the DSP subsystem, the DSP MMU, and the DSP interrupt modules.

At reset, the CK_GEN2 is in the bypass mode, so it supplies (CLKM2) a clock of the same 12- or 13-MHz frequency as the CLKIN. After the global-reset period, the MPU application program can change the clock frequency through the CK_GEN2 control register.

The DSP_CK is enabled at reset until the DSP is in the reset state. The EN_DSPCK bit (located in the clock-control register ARM_CKCTL) allows the MPU to turn off the DSP_CK while the DSP is held in a reset state.

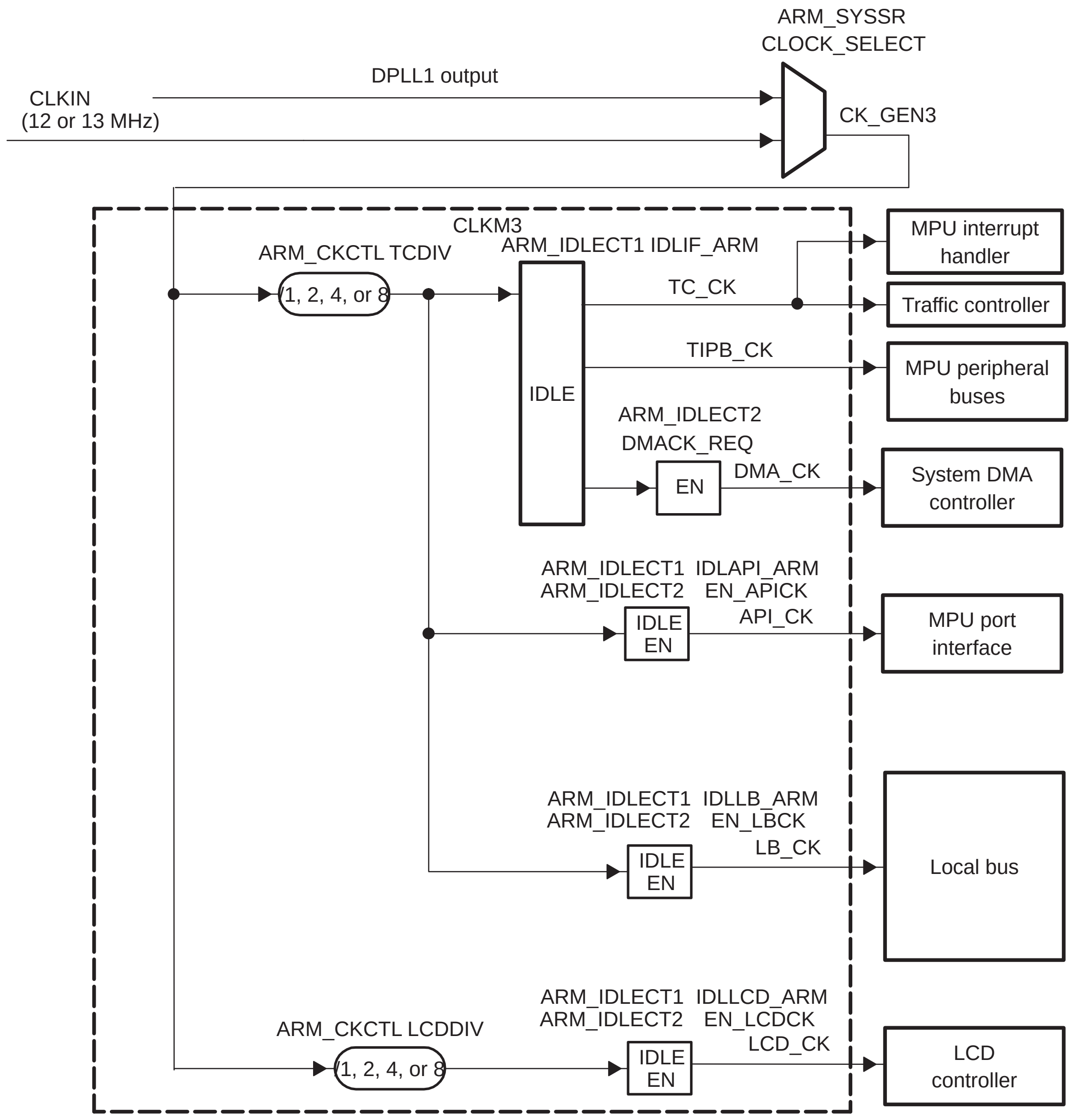
A free-running counter/divider receives the CK_GEN2 signal and makes available four taps where a 50%-duty-cycle clock and the clock divided by 1, 2, 4, and 8 can be selected. A multiplexer set, through the DSPDIV bits in the clock-control register (ARM_CKCTL), selects the clock frequency that applies to the DSP clock domain.

At reset, the higher frequency (divide by 1) is selected. The software-application program (accessing the control-register file) can change the divider ratio by writing to the DSPDIV [1–0] bits at any time while the OMAP5910 device is running. A synchronization mechanism is implemented to remove any spikes while the clock frequency is changing (disable the clock first, change the DSPDIV bits, and then enable the clock).

2.6 CLKM3

The CLKM3 controls the clock-distribution and idle modes of the traffic controller and various system-level-clock domains. The traffic-controller clock, CLKM3 (see Figure 7), has the following domains.

Figure 7. Traffic Controller Clock Distribution



Many of the following clocks are the same as the traffic-controller clock (TC_CK) in terms of their frequencies, but not their IDLE controls. Each of the clocks has separate IDLE control logic.

❑ The traffic-controller clock, TC_CK, is derived from CK_GEN3 divided by 1, 2, 4, or 8, as programmed via the TCDIV bits of the MPU-clock-control register (ARM_CKCTL). The MPU interrupt handler uses the TC_CK clock, as set by the ARM_INTHCK_SEL bit of the MPU-clock-control register (ARM_CKCTL). The idle mode is controlled by the IDLIF_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1).

- The local-bus and local-bus-MMU clock, LB_CK, is the same as the TC_CK. The idle mode is controlled by the IDLLB_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1). The LB_CK is enabled by the EN_LBCK bit of the MPU-idle-mode-entry-2 register (ARM_IDLECT2).
- The MPU-port-interface (MPUI) clock is dependent not only on the IDLAPI_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1), but also on DSP_IDLE.
- The system-DMA-controller clock, DMA_CK, is the same as the TC_CK. The idle mode is controlled by the IDLIF_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1), and the clock is enabled by the DMACK_REQ bit of the MPU-idle-mode-entry-2 register (ARM_IDLECT2).
- The MPU-peripheral-bridge clock, TIPB_CK, is the same as TC_CK. The idle mode is controlled by the IDLIF_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1).
- The LCD-controller clock, LCD_CK, is derived from CK_GEN3 divided by 1, 2, 4, or 8, as programmed via the LCDDIV bit of the MPU-clock-control register (ARM_CKCTL). This clock is enabled by the EN_LCDCK bit of the MPU-idle-mode-entry-2 register (ARM_IDLECT2). The idle mode is controlled by the IDLLCD_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1).

2.7 Clock Distribution and Synchronization

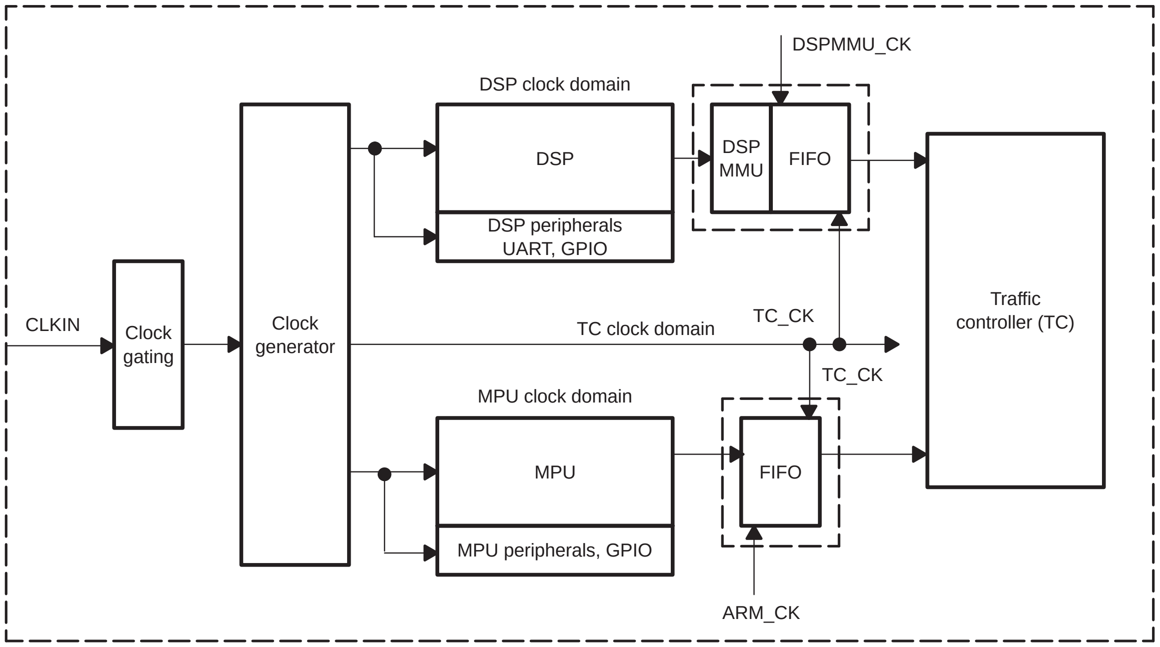
When any of the clock domains are running at different frequencies with respect to each other, the FIFOs are used to buffer data being transferred between domains (see Figure 8). This is necessary to ensure that data being sent from a faster domain is buffered as a slower domain receives it. This buffering introduces latencies as data is passed between the domains. Thus, this buffering can be bypassed if it is not needed (that is, when the domains are running at the same speeds).

- For the fully-synchronous-clocking scheme, ARM_CK = DSPMMU_CK = TC_CK; the FIFO logic is bypassed between the TC and MPU, and the TC and DSPMMU.
- For the synchronous-scalable-clocking scheme, the FIFO logic is used for both processors.

Note:

The TC_CK clock must be slower than or equal to the ARM_CK and DSP MMU clock speed.

Figure 8. OMAP5910 Clock Distribution and Synchronization

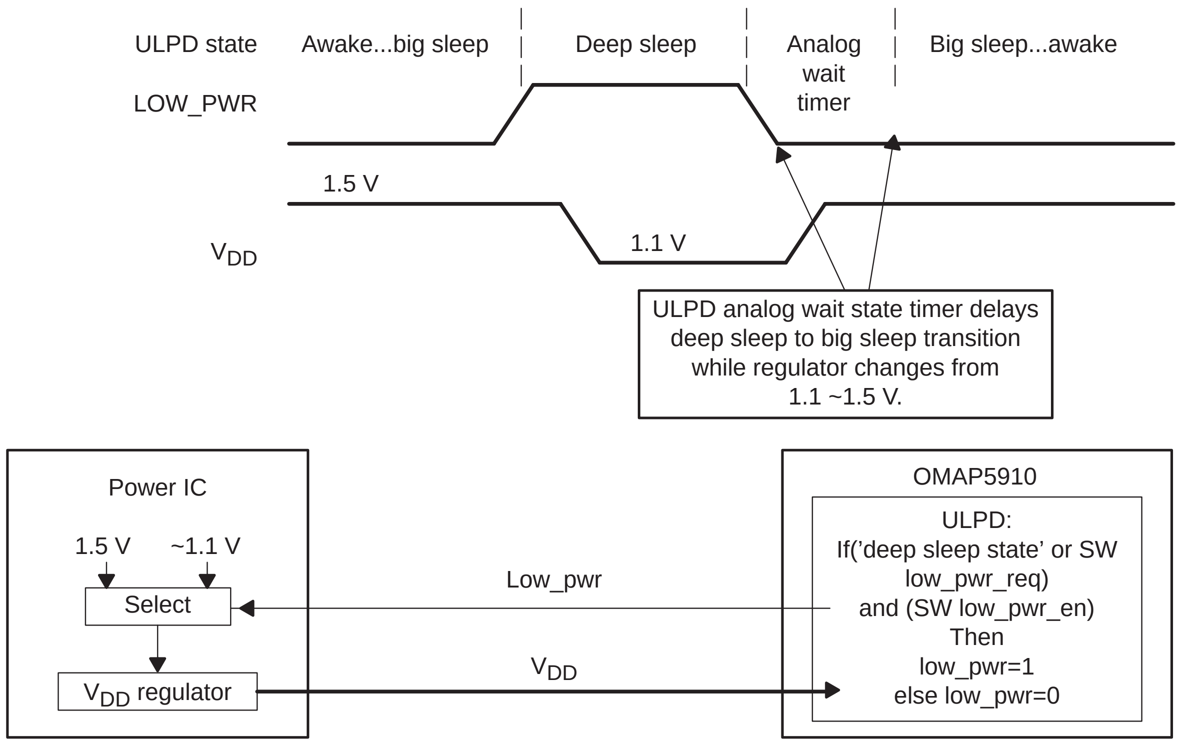


2.8 Low-Power Mode

The OMAP5910 LOW_PWR I/O is an active-high request for the LOW_PWR mode (see Figure 9). It automatically requests external regulators to go to standby (low-power mode) or to lower the VDD core voltage during the deep-sleep mode. A software request, through bit 1 of the power-control register (POWER_CTRL_REG), is available for debugging and future support of the low-voltage operational mode.

The LOW_PWR signal is multiplexed on the MPUIO5 ball. To get the signal on this ball, bits [14:12] of the FUNC_MUX_CTRL_7 register must be set to 001.

Figure 9. Low-Voltage Mode



3 Power Management

There are three clock domains in the OMAP5910 device. Each clock domain has its own clock-management unit and can be put into idle mode (power-saving mode) when unused without affecting the rest of OMAP5910 functionality. In addition, the ULPD provides the low-power modes that affect the entire device, not just the individual domains (see Figure 10).

A chip idle occurs when the DSP is idled, the MPU requests an idle, and the TC domain has no remaining transactions. A chip idle causes the ULPD to initiate the big-sleep mode. In the big-sleep mode, the DPLL 1 is turned off, but the 12-MHz clock is still active. If the 12-MHz clock is not needed, then the ULPD can initiate a further transition to the deep-sleep mode, which turns the 12-MHz clock off internally.

When an unmasked-interrupt event occurs, a request is performed by the wake-up-control module. When receiving the request, if the device is in deep-sleep mode, the ULPD brings the device out of the deep-sleep mode. Once the 12-MHz clock is stable, the ULPD brings the device into the awake mode.

To reduce the wake-up time, a special hardware request is implemented in parallel to interrupts to wake-up the ULPD whenever an interrupt occurs. This request is generated by peripherals such as the USB or UART. When receiving the hardware request, the ULPD brings the device out of the deep-sleep mode to wake-up the 12-MHz clock. When the clock is awake, the ULPD goes into the big-sleep mode and awakes if a request is received from the wake-up-control module. Figure 11 shows the wake-up-control module.

The deep-sleep mode is the entry state of the device when a power-on reset occurs. Such a reset acts as a wake-up request, causing the device to transition to the awake state.

The 12-MHz clock may be required to clock signals out of the device (such as the MCLK pin) or to the ULPD DPLL (used for the USB and other internal peripherals) in either the big-sleep or awake states. The deep-sleep state can not be entered if there is need for the 12-MHz clock.

The power-management state machine runs at 32 kHz. All control signals of this state machine are resynchronized on the 32-kHz clock. The 32-kHz clock is always on.

Table 3 lists the peripherals and external signals which can wake up the OMAP5910 device from the deep-sleep mode.

Figure 10. Power Management State Machine

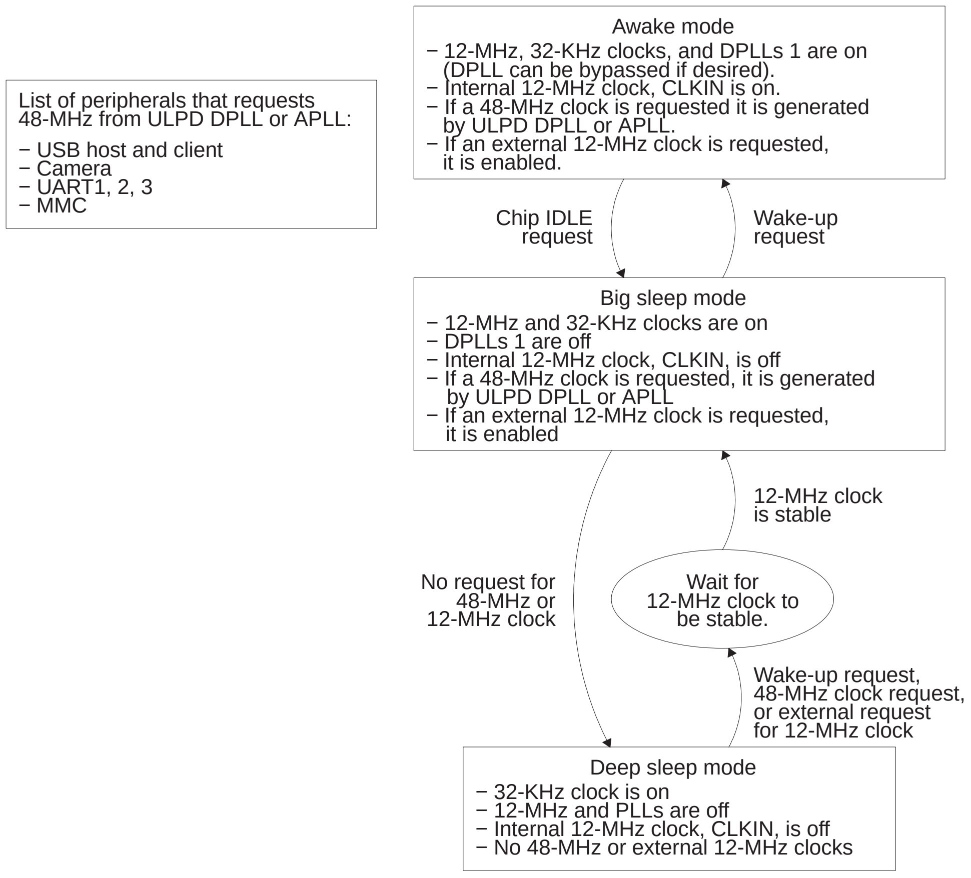

List of peripherals that requests 48-MHz from ULPD DPLL or APLL:
− USB host and client
− Camera
− UART1, 2, 3
− MMC
Awake mode
− 12-MHz, 32-KHz clocks, and DPLLs 1 are on (DPLL can be bypassed if desired).
− Internal 12-MHz clock, CLKIN is on.
− If a 48-MHz clock is requested it is generated by ULPD DPLL or APLL.
− If an external 12-MHz clock is requested, it is enabled.
Chip IDLE request
Wake-up request
Big sleep mode
− 12-MHz and 32-KHz clocks are on
− DPLLs 1 are off
− Internal 12-MHz clock, CLKIN, is off
− If a 48-MHz clock is requested, it is generated by ULPD DPLL or APLL
− If an external 12-MHz clock is requested, it is enabled
12-MHz clock is stable
No request for 48-MHz or 12-MHz clock
Wait for 12-MHz clock to be stable.
Wake-up request, 48-MHz clock request, or external request for 12-MHz clock
Deep sleep mode
− 32-KHz clock is on
− 12-MHz and PLLs are off
− Internal 12-MHz clock, CLKIN, is off
− No 48-MHz or external 12-MHz clocks

Figure 11. Wake-up Control Module

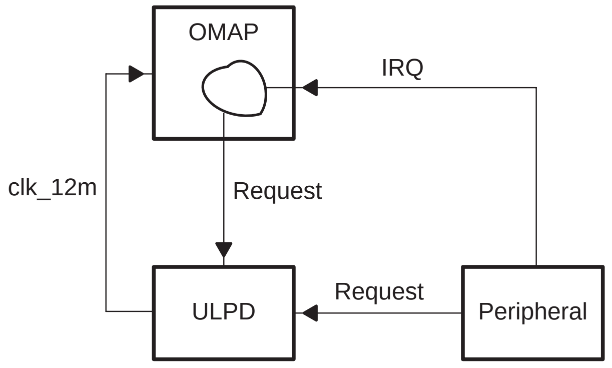


Table 3. OMAP5910 Wake-Up Peripherals and External Signals

Requestor	Mechanism	Transition from Deep Sleep to ?
Power on reset (external signal)	Cold reset	AWAKE
$\overline{\text{MPU\_RESET}}$ (external signal)	Warm reset	AWAKE
MPUIO keyboard	MPU interrupt	AWAKE
MPUIO GPIO	MPU interrupt	AWAKE
Timer32K	MPU interrupt	AWAKE
UART2 RX detection	Peripheral request generated to ULPD	AWAKE
RTC	MPU interrupt	AWAKE
USB cable insertion	USB request to ULPD which generates MPU interrupt	BIG SLEEP then AWAKE via the MPU interrupt
UART1/2/3 (wait for a falling edge on the RX, DSR or CTS signals)	MPU/DSP interrupt	AWAKE
MCLK_REQ (external signal)	Request to ULPD	BIG SLEEP
BCLK_REQ (external signal)	Request to ULPD	BIG SLEEP
External DMA request (external signal)	Asynchronous request from TC	AWAKE

3.1 DSP-Idle Modes

Two DSP registers are used to configure and check the idle modes for the DSP subdomains.

- ❑ The idle-configuration register (ICR) specifies which clock domains get put into idle by the next execution of the IDLE instruction.
- ❑ The idle-status register (ISR) indicates which subdomains are currently in IDLE mode.

The six different subdomains are:

- ❑ The DSP-core subdomain (DSP CORE/SARAM/DARAM): ICR and ISR bit 0
- ❑ The DSP-DMA-controller subdomain (DMA/SARAM/DARAM): ICR and ISR bit 1
- ❑ The I-cache subdomain (I-cache): ICR and ISR bit 2

- The Peripherals subdomain (peripherals outside the DSP): ICR and ISR bit 3:

  Peripherals can be individually controlled (shut off/enabled) by programming the control registers in the clock-generation-management module (CLKM). To maximize power conservation, seven different peripheral-idle modes are defined (UART, GPIO, timers, watchdog timer). Each one can be individually activated and deactivated by software.

  Two different strategies are used to control the clock that feeds the DSP peripherals:

  - The clock is shut off/activated according to the DSP-idle mode or the application-specific environment (disable the peripheral clocks when the DSP in not in idle). Peripherals connected to this clock cannot request DMA transfers during the DSP-idle mode.
  - The clock is never shut off (input reference clock).

    In either case, the DSP peripheral clocks are directly shut off/activated by the DSP software.

- The DPLL subdomain (DSP input clock + DSP interrupt handler): ICR and ISR bit 4

  Setting up the DSP-DPLL-idle mode sends a DSP_IDLE signal to the clock-generation-management module (CLKM), which disables the input clock to the DSP.

- The DSP-EMIF subdomain: ICR and ISR bit 5

3.1.1 Putting the DSP in IDLE

Perform this procedure to put the DSP in idle mode:

1) Turn off the DSP-peripheral clocks.
2) Disable the DSP-watchdog timer.
3) Enable the desired DSP interrupts (interrupt mask).
4) Enable the DSP-global interrupt (INTM bit), if required.
5) Switch the DSP to shared-access mode (SAM) (it is in the SAM mode, but switch the mode to signal the CPU). In the SAM mode, the public peripheral bus is shared between the MPU and DSP.
6) Write to the ICR registers to disable all clock domains or a particular clock domain (write a 1 to disable).

7) Switch the DSP back to host-only mode (HOM). In the HOM mode, the public-peripheral bus is owned exclusively by the MPU (usually because the DSP is in idle mode or about to go to idle mode).

8) Execute the IDLE instruction.

   The DSP goes to sleep. If INTM is set, the ISR is not executed after wake up; the interrupt simply wakes the DSP up. The program continues just after the IDLE instruction; otherwise, the ISR is executed.

When the DSP is awakened (by an enabled interrupt or an external event such as reset), perform the following procedure:

1) Disable the global interrupt (INTM), if required.

2) Switch the DSP to the SAM mode. The public-peripheral bus is now shared by the DSP and MPU.

3) Write a 0 to the respective ICR bit to clear the idle bit.

4) Execute the IDLE instruction.

Note:

To set the DPLL subdomain to idle, switch off the other clock domains (CPU, DMA cache); otherwise, the DSP peripheral cancels the idle request with a bus error indicating that the configuration is not allowed.

3.2 MPU-Idle Modes

A clock-management register, the MPU-idle-mode-entry-1 register (ARM_IDLECT1), controls the different clock domains (clock enables) during the idle state and allows the user to put different parts of the MPU clock domain into the idle mode, if desired.

Three different subdomains are defined: the MPU subdomain, the DPLL subdomain, and the MPU peripheral subdomain.

3.2.1 MPU Subdomain (MPU + MPU Interrupt Handler)

The MPU can go into the idle mode in two ways:

❑ By executing the CP15 , wait-for-interrupt, instruction*:* Executed by an OS kernel. By configuration, this instruction provokes an Idle1 or an Idle2 mode.

❑ By setting the SETARM_IDLE bit (ARM_IDLECT1[11], (the idle-control register in the clock-generator and system-reset module): Programmed by a process application. Setting these bits (active) allows the MPU to enter in an Idle1 or an Idle2 mode. This is the recommended method to use.

The MPU clock restarts upon an enabled-interrupt request or system reset.

Wait-For-Interrupt Instruction

When the MPU CP15 instruction is used, the system software does not need to take care of adding any extra cycles to wait after the instruction being executed (the MPU itself takes care of this). When this instruction is executed, the MPU stops its ongoing operations and sends a sleep acknowledge signal to the OMAP5910 clock reset module to request stopping the clock. The MPU does not execute any other access after executing the wait-for-interrupt instruction. The MPU wakes up when an interrupt occurs and executes the subsequent instructions after servicing the interrupt.

Set SETARM_IDLE (ARM_IDLECT1[11])

After SETARM_IDLE is set through software, the software must wait for a certain number of cycles to ensure no other MPU requests occur before the MPU is idled. This is because there is a latency between the MPU-TIPB write to the IDLECT1 register and the time when the MPU-sleep request to the MPU core actually goes high. Also, the software must account for the clock cycles required for the MPU to send an acknowledge signal back to the OMAP clock reset module, depending on what operations it was performing.

The following rules ensure that sufficient time is allowed:

- The clock-reset module needs at least four reference-clock cycles (12-MHz clock) and two MPU-clock cycles to send a sleep request to the MPU.
- The MPU needs at least three MPU-clock cycles to send back an acknowledge after receiving the sleep request. This is true even if there are no access requests coming from the MPU before or after the sleep request. The sleep request is generated from the register write to the IDLECT1 Bit 11.
- Add NOP instructions to make sure there is no request coming from the MPU by considering the worst-case scenario. This scenario is: MMU and I-cache are enabled, a MMU-TLB miss (requiring a L1 and L2 fetch), and an I-cache miss, as follows:
  - Four read strobes for the instruction fetch (a line load of 4 words)
  - One read strobe for the TLB-Miss-L1-descriptor fetch
  - One read strobe for the TLB-Miss-L2-descriptor fetch

This requires six read strobes, which need N number of MPU clock cycles. N depends on the type of memory from which the reads are being made.

❑ There is a software solution to avoid these extra N MPU-clock cycles due to the read strobes. This solution requires that only I-Cache be enabled. By changing the loop-count value (CMP R2), the number of cycles can be increased/decreased, as shown in Figure 12.

Figure 12. Code Example

```
    .state16            ; thumb mode
    .ref   edata        ; defined by armas
    .global $arm_idle
$arm_idle:
    push   {lr}
    push   {r1–r7}
    adr    r4, into_32_bis
    bx     r
    nop
    nop
    nop
    .state32            ; arm mode
into_32_bis:
    LDR    R1,ARM_IDLECT1
    MOV    R3,#1
    MOV    R3,R3,LSL # 11
    MOV    R2,#0
    LDR    R0,[R1]
    ORR    R0,R0,R3
; This is the loop that will wait for at least 100 cycles
; before issuing next request from ARM. On the first run of the loop only Icache
; gets loaded with the loop and the next 2 instructions but write to SYSST does
not occur
; In the 2nd run of the loop only write to IDLE_CT1 happens and after that ARM
runs the loop from
;Icache so no request goes out
LOOP   CMP    R2,#1
       STREQ  R0,[R1]
       ADD    R2,R2,#1
       CMP    R2,#16
       BNE    LOOP
```

Figure 12. Code Example (Continued)

```
the_end:
    adr r2, into_16_bis + 1
    bx  r2
    .state16
into_16_bis:
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    pop     {r1–r7}
    pop     {pc}
;
CONSTANT TABLE
;
ARM_IDLECT1                .long 0xFFFECE04
```

Example:

❑ If the reference-clock speed = 12 MHz, then the MPU-clock speed = 96 MHz

❑ If the above routine is used after the write to the ARM_IDLECT1 register, 4 * 96/12 + 2 + 3 = 37 clock cycles are needed

To achieve this, run the loop seven times.

3.2.2 DPLL Subdomain

The DPLL1 is disabled when all of the clock domains using the DPLL1 are disabled.

Setting up the IDLDPLL_ARM bit enables the DPLL1 to enter the idle mode when the following conditions are met; otherwise, the DPLL1 remains active.

- The DSP is set in the global-idle mode.
- The MPU is set in the idle mode (either from a request or from a wait-for-interrupt instruction).
- There is no active DMA transaction and there is no activity on the internal local bus, as indicated by the internal TCLB_EN signal
- No peripheral post write is queued.
- The peripheral clocks are stopped.
- There are no pending interrupts.

3.2.3 Peripheral Subdomain

Two clocks feed the MPU peripherals:

- The clock that is shut off/activated according to the MPU-idle mode. Peripherals connected to this clock cannot request DMA transfers during the MPU-idle mode.
- The clock that is never shut off (input reference clock)

In either case, the MPU peripheral clocks are directly shut off/activated by the MPU software.

3.3 Traffic-Controller-Idle Modes

A clock-management register, ARM_IDLECT1, controls the different clock subdomains (clock enables) during the idle state and allows the user to put different parts of the traffic controller into idle mode, if desired.

Several different subdomains are defined.

- System-DMA subdomain

  To optimize power consumption, the system-DMA-controller clock (DMA_CK) can be shut off/activated according to the DMA activity (provides the clock during data transfer only) or it can be shut off only when the MPU goes to idle.
- Traffic-controller subdomain

  The traffic-controller clock (TC_CK) is shut off if the MPU and the DSP DPLLs are in idle mode and there is no DMA transfer. When the clock must be shut off, the memory interface completes the current memory transaction before notifying the CLKM3 to shut off a clock. The SDRAM is placed in self-refresh mode before shutting off the SDCLK_EN clock, and there is no internal local-bus activity, as indicated by the internal TCLB_EN signal.

- MPU-peripheral-bridge domain

  The MPU-peripheral-bridge clock (TIPB_CK) is shut off if the MPU is in the idle mode and there is no DMA transfer.

- LCD domain

  The LCD clock (LCD_CK) is shut off/activated according to the traffic-controller-idle mode or forced off with the enable bit.

- MPU-MPUI domain

  The MPUI clock (API_CK) is shut off if the MPU and the DSP DPLLs are in the idle mode and there is no DMA transfer (when the clock is required to be shut off, the memory interface completes the current memory transaction before notifying the CLKM3 to shut off the clock). The MPUI clock can also be forced off with the enable bit (the MPUI clock can only be shut off by the enable bit in current implementation).

Note:

To idle the traffic controller, the system software must ensure that the current transfers are completed and their clock domains are turned off before going to idle.

The traffic-controller clock restarts upon either an MPU or DSP interrupt request, a DMA request, or activity on the internal buses.

3.3.1 DPLL-Idle Procedure

In the event that only the input-reference clock is needed (that is, only timer/watchdog are active), then the DPLL can be set to idle mode. This procedure applies to DPLL. The DPLL clock is stopped if all the domains that use it as a source are stopped.

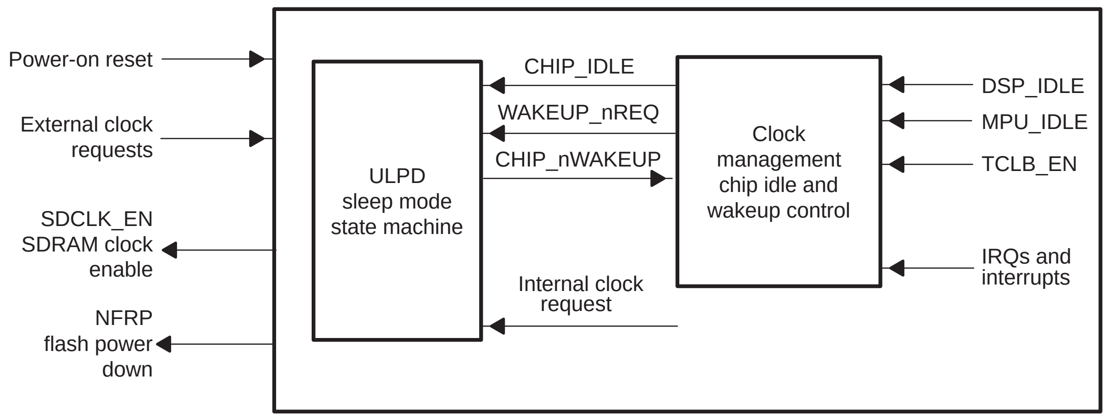
Setting the IDLDPLL_ARM bits (in ARM_IDLECT1 register) to logic 1 forces the corresponding DPLL to enter the idle mode whenever the DPLL output clocks *(CK_GEN1, CK_GEN2, CK_GEN3)* are not being used.

The DPLL-idle control logic receives signals from clock-generation modules, indicating when the output clock can be stopped. The DPLL-idle mode is entered when the IDLDPLL_ARM bits in the MPU-idle-mode-entry-1 register (ARM_IDLECT1) are set to logic 1.

3.4 Chip Idle and Wake-Up Control

In the status-request register (STATUS_REQ_REG), both the CHIP_IDLE and WAKEUP_nREQ signals can be used by the ULPD idle logic to control the input-clock source. The CHIP_IDLE signal remains high until the DPLLs acknowledge that they have entered the IDLE mode. The WAKEUP_nREQ signal goes active as soon as one of the chip-wake-up conditions occurs (MPU or DSP interrupt, DMA request, or local-bus activity). The CHIP_IDLE signal can be used by the ULPD to decide when to stop the CLKIN (CLKIN) to the clock-generation module, and it uses the WAKEUP_nREQ signal to restart the clock (see Figure 13).

Figure 13. Chip Idle and Wake-Up Control



In the status-request register (STATUS_REQ_REG), the following signals are used in chip idle and wake-up control:

- ❑ CHIP_nWAKEUP: CHIP_nWAKEUP is the acknowledge of the ULPD when the CHIP_IDLE signal is active. It indicates to the clock-management module that the ULPD has left the awake mode and the 12-MHz clock, CLKIN, is off.

- ❑ When CHIP_IDLE becomes inactive, the ULPD releases the CHIP_nWAKEUP signal to indicate that the ULPD is in the awake mode and the CLKIN is back on. This signal provides the ULPD with more control of the OMAP5910 device wake-up.

- ❑ When the MPU-idle-mode-entry-1 register (ARM_IDLECT1) WKUP_ MODE field = 0, the OMAP5910 device does not wake up, even if one of the above wake-up conditions occurs, until the CHIP_nWAKEUP goes active.

- MPU interrupt
- DSP interrupt
- Internal local-bus activity (TCLB_EN signal = 1)

In the chip-idle mode, if this signal is inactive (high), the OMAP5910 device does not wake up (prevent from waking up), even if one of the above wake-up conditions has occurred. This signal has no effect when the OMAP5910 device is not in the chip-idle mode. When the MPU-idle-mode-entry-1 register (ARM_IDLECT1) WKUP_MODE = 0, the OMAP5910 device does not wake up, even if one of the above wake-up conditions has occurred.

- WAKEUP_nREQ: request to the ULPD to wake up the chip. This signal is looked at only when the CHIP_IDLE signal is high. Note that WAKEUP_nREQ is asserted by the OMAP5910 device whenever any wake-up condition occurs, even when the OMAP5910 device is not in CHIP_IDLE.
- $\overline{\text{FLASH.RP}}$ and SDRAM.CKE: Power control for external devices (for example, flash and SDRAM)
- Power-on reset: Must be valid until the power and input clocks are stable
- CHIP_IDLE: Chip-idle mode. Indicates that all internal clocks are stopped. This internal signal is active regardless of the state of the external wake-up-control feature. This pin is asserted (high) when all DPLL ACKs are returned and all the peripherals using CLKIN are disabled.
- TCLB_EN: TC local bus enable, enables the restart of the clock to the TC when the idle mode is set.
- DSP interrupts (internal signal is DSP_nIRQ): Interrupt request from the DSP-interrupt handler. Asserts when an unmasked interrupt has been asserted.
- MPU IRQ (internal signal is nIRQ_SET*):* Interrupt request from the MPU-interrupt handler. Asserts when an unmasked interrupt has been asserted. When the MPU is awake, must remain low until the MPU clock restarts.
- DSP_IDLE—DSP-idle command
- MPU_IDLE—MPU-idle command

3.4.1 Chip-Idle Mode

In the event that no clock signal is necessary (chip idle), all clock domains generated either from the CK_GEN (1,2,3) or CLKIN are stopped. The external reference-clock source can be stopped as well.

The CHIP_IDLE signal is asserted when all internal system clocks are disabled and after the DPLL idle state has been acknowledged. It is deasserted when a wake-up condition is detected (CHIP_nWAKEUP and one of the wake-up conditions is active when the external wake-up control is enabled or just one of the wake-up conditions is active when the wake-up control is not enabled). It takes some synchronization CLKINs for the CHIP_IDLE to go low after a wake-up condition is detected.

Note:

In addition to the DPLL timing, the clock source start-up time can affect the OMAP5910 system response. Use the deep-sleep mode for long sleep-periods only.

3.4.2 Chip-Idle Procedure

1) Set TC_EMIF_SLOW_IF_CONFIG_REG = 0x0000000C and TC_EMIF_FAST_SDRAM_CONFIG_REG = 0x0C618800

2) Disable the MPU watchdog timer by first writing 0x00f5 to the WDTIMER_TIMER_MODE_REG and then writing 0x00A0 (this is to prevent a watchdog reset from being generated that results in a global reset.)

3) Set up the MPU interface (write to the MPUI-control-setup and DSP-boot registers).

4) Set API_SIZE_REG = 0x0000 (only need to set the bits that correspond to the SARAMs with DSP code to 0).

5) Enable the MPU interrupts and unmask these interrupts by writing a 0 to the corresponding bit in the MIR-mask-interrupt register in the MPU-interrupt handler. Write to the corresponding interrupt-level register (ILR) to set the priority, edge sensitivity, and whether the interrupt is routed to the IRQ or FIQ. Besides the interrupt that is used to wake up the MPU out of idle, a DSP-mailbox interrupt to the MPU can also be enabled. This DSP2MPU-mailbox interrupt service routine is used to put the MPU into idle. The reason to use this mailbox interrupt is that the MPUI clock is needed by the DSP to switch between the SAM/HOM modes. The MPU

cannot go to idle until this is done, because only the MPU can write to the EN_APICK bit of the MPU-idle-mode-entry-2 register (ARM_IDLECT2).

6) Take the DSP out of reset. First, write the MPU-reset-control-1 register (ARM_RSTCT1) DSP_RST field = 1. Set the MPU-reset-control-1 register (ARM_RSTCT1) DSP_EN to = 1. The DSP_RST bit controls the MCU reset, and the DSP_EN bit controls the reset signal of the DSP.

7) In the DSP code, enable and unmask the DSP interrupts.

8) Disable the DSP-watchdog timer (that is, take it out of the watchdog mode) and disable the DSP-peripheral clocks setting the DSP-idle-mode-entry-2 register (DSP_IDLECT2) to 0x0000.

9) Write to the DSP-mailbox registers to generate the interrupt to the MPU. In the corresponding MPU-interrupt service routine, add a wait loop to make sure the DSP has gone to idle before disabling the MPUI clock.

10) In the MPU-interrupt service routine, clear the DSP2MPU-mailbox interrupt, then disable the ARMGPIO_CK, LB_CK, and LCD_CK by writing the MPU-idle-mode-entry-2 register (ARM_IDLECT2) = 0x0087 (can also write 0x0000, which disables the MPU peripheral clocks instead of letting them go to IDLE, only after the MPU goes to IDLE using the MPU-idle-mode-entry-1 register (ARM_IDLECT1) IDL_ARM). Put the MPU into idle mode by writing the MPU-idle-mode-entry-1 register (ARM_IDLECT1) = 0x0FFF, which sets the SETARM_IDLE bit. This also sets the IDLIF bits, which allow the MPU peripherals to go to idle when the MPU goes to idle, and sets the IDLDPLL_ARM bit, which allows the DPLLs to go to idle.

11) Back in the DSP code, after writing to the DSP-mailbox register in step 10, switch the DSP TIPB and MPUI to the SAM mode. Then, write 0xFF to the DSP ICR register. Switch the DSP back to the HOM mode and execute the IDLE instruction. See the section on putting the DSP in idle mode for descriptions of the SAM and HOM modes.

    The DSP and MPU domains go to idle, and then the TC also goes to idle. Then all 3 DPLLs go to idle, and the CHIP_IDLE signal goes active high, which indicates to the external system that the OMAP5910 input clock can be disabled, assuming there are no wake-up conditions at that time indicated by the WAKEUP_nREQ signal.

3.4.3 Wake-Up Procedure

An interrupt request (not masked) (either to the MPU or to the DSP), a DMA clock request, or a logical 1 at the TCLB_EN signal exits the idle mode. In addition, when all internal clocks are stopped (chip-idle mode), the wake-up procedure can be controlled via the ULPD. The WKUP_MODE bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1) defines this wake-up option.

To give the ULPD wake-up control, the WKUP_MODE bit must be cleared to 0 before entering the idle mode.

In the OMAP5910 device, the WKUP_MODE bit is controlled by the MPU in the MPU-idle-mode-entry-1 register (ARM_IDLECT1). The DSP has no control of this bit.

When the WKUP_MODE-bit value is set to logic 1, a single wake-up condition (as defined in the following list) initiates a chip wake-up procedure.

1) nIRQ_SET: Upon an interrupt request, the MPU-interrupt handler initiates the restart of the ARM_CK, ARM_INTH_CK, TIPB_CKs, DMA_CK, and TC_CK clocks (depending on the setting of the MPU-idle-mode-entry- 1/2 registers (ARM_IDLECT1/2), peripherals clocks can also restart). If the idle mode was entered from the SETARM_IDLE bit, then the bit is cleared to 0.

2) DSP_nIRQ: Upon an interrupt request, the DSP-interrupt handler initiates the restarting of the MPUI clock (if MPUICK_EN is not set to 0), DSP_CK, DSP_INTH_CK, TC_CK clocks (depending on the setting of the MPU-idle-mode-entry-1/2 registers (ARM_IDLECT1/2), peripheral clocks can also restart).

3) TCLB_EN signal: When the internal TCLB_EN signal goes active, the TC_CK and LB_CK clocks restart. The TC_CK/LB_CK clocks keep running as long as the local-bus activity occurs.

4) When the system-DMA controller receives an asynchronous request from the traffic controller, the DMA_CK/TC_CK/LB_CK and DMA_CK/TC_CK/LB_CK are enabled to keep running as long as the DMA operates.

5) When the system-DMA controller receives a request from the TIPB bridge, it enables the TC_CK/TIPB_CKs/DMA_CK and TC_CK/TIPB_CKs/D MA_CK to keep running as long as the DMA operates.

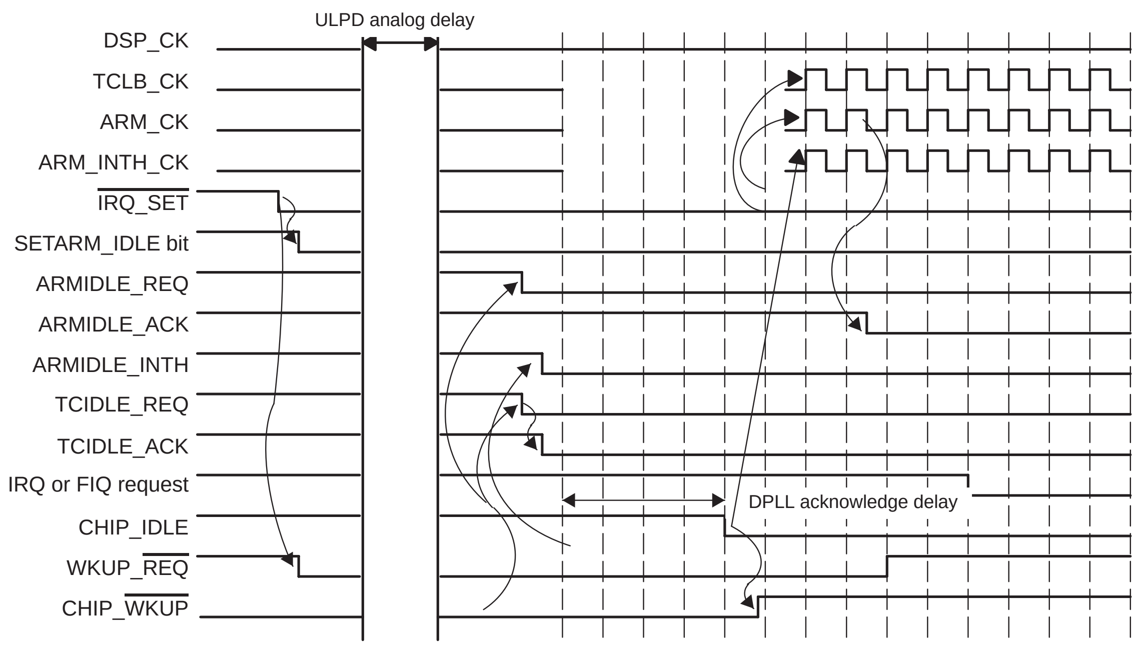
Note:

The internal signals that cause the wake-ups are asynchronous and do not need a running clock to be activated. When the WKUP_MODE bit value is a logical 0 and the CHIP_IDLE signal is active, this condition indicates the entire chip is in the deep-sleep mode. The combination of one of the above conditions and a CHIP_nWAKEUP request from the ULPD is required to exit the chip-idle mode. If WKUP_MODE = 1, then the ULPD is not required to exit the chip-idle mode, only one of the above wake-up conditions.

If the CHIP_IDLE signal is inactive (and at least one of the internal clocks is running), the CHIP_nWAKEUP signal is disabled and a single wake-up condition (not ULPD controlled) brings the DSP or MPU system out of idle mode. A global-system reset brings the OMAP5910 device out of the idle mode, regardless of the WAKEUP_MODE bit value, ULPD control, or the interrupt status.

Figure 14 illustrates a ULPD-controlled wake-up sequence (assuming the DSP and the MPU clock domains have the same clock frequency, CLKIN). The wake-up is initiated from an interrupt to the MPU, whereas the DSP remains in the idle mode.

Figure 14. ULPD Controlled Wake-Up Sequences



3.5 Power-Saving Capability

The OMAP5910 device has several power-saving modes that help to reduce the operating current by stopping the clock signals of the unused (inactive) domain(s). The idle controls to the DSP, the MPU, and the traffic controller provide a flexible and an efficient power-saving mechanism. The following list of power-saving modes is given as an example only, because other modes are possible. The system software can program the OMAP5910 device to operate in any of these modes for a specific application.

- Mode 0: The DSP is partially in the idle mode (see the DSP idle protocol).
- Mode 1: The DSP is in the global-idle mode.
- Mode 2: The MPU is in the idle mode (the DSP is still running) for both:
  - System DMA controller is active.
  - System DMA controller is not active.
- Mode 3: Both the MPU and DSP are set in the idle mode (peripherals are still active) for both:
  - System DMA controller is active.
  - System DMA controller is not active.
- Mode 4: The MPU, the DSP, and traffic controller are stopped (the traffic controller can be stopped only if both the MPU and the DSP are in idle) for both:
  - Peripheral modules are individually stopped.
  - All peripherals are stopped.
- Mode 5: The MPU, the DSP, the peripherals, and the DPLL (1) are stopped; however, the timer/watchdog (or OS-timer) is still active.
- Mode 6 (chip idle): The MPU, the DSP, the peripherals, the DPLL (1), and timers are stopped, while the ULPD clock source remains the only active clock signal.
- Mode 7 (deep sleep): All internal system clocks (the MPU, the DSP, the peripherals, the DPLL (1) and the timers) and the ULPD reference-clock source are stopped, leaving the OMAP5910 device in a static state in which it consumes the lowest-possible power.

In all power-saving modes, the OMAP5910 device retains all RAM data (keeps the memory data), and the register configuration values (for example, the frequency selection is maintained, etc.). The data to output terminals is also maintained, and input terminals are set to logic low or logic high (not floating) to reduce the current from flowing through the input logic.

The OMAP5910 device exits from the power-saving modes by means of a reset or any interrupt (with or without additional external control from the CHIP_nWAKEUP pin). A wake-up interrupt must be enabled (not masked off) to bring the OMAP5910 device out of the power-saving mode.

3.6 ULPD Power-Management State Machine

The power-management function of the ultra low-power module (ULPD), handles the high-frequency oscillator on/off sequences. It is a state machine that can stop the oscillator and restart it on a wake-up signal. Set-up times of the oscillator are taken into account in order to stop/restart internal clocks in a clean manner. The ULPD state-machine uses the 32-kHz clock.

- The ULPD DPLL and APLL for the 48-MHz USB clock are handled in the ULPD module.
- ULPD DPLL is a x4 digital PLL.
- APLL is a x48 analog PLL. The input clock reference is 1 MHz, based on either a 12-MHz system clock divided by 12 or optionally on a 13-MHz system clock divided by 13.
- The switch between the DPLL and APLL is controlled by software through a TIPB register of the ULPD.

3.6.1 Gauging the 32-kHz Oscillator

As the 32-kHz oscillator exact frequency is unknown, it can be gauged by comparing the 32-kHz oscillator with a high-frequency clock (12-MHz oscillator, ULPD DPLL, or external clock) during any active period. Gauging is only necessary in specific applications where it is important to know the exact frequency of the 32-kHz oscillator.

There is a software limitation: the counter is not resynchronized on the TIPB strobe. Therefore, the value is not readable while the counter is running (when gauging is enabled). The gauging (GAUGING_CTRL_REG[0] = 0) must first be disabled, then read the high-frequency counter and the 32-kHz counter value.

3.6.2 Gauging Versus the High-Frequency Clock

To gauge the 32-kHz oscillator, two counters clocked on the 32-kHz clock and the high-frequency clock are concurrently running during the gauging period. The high-frequency clock is selected among the 12-MHz clock, the DPLL, and the external clock. At the end of the gauging period, the number of 32-kHz clocks and the number of high-frequency clocks are calculated as follows:

- ❑ Nb_32kHz = counter_32_msb * 65536 + counter_32_lsb:
  - counter_32_msb is the MSB value of 32-kHz counter.
  - counter_32_lsb is the LSB value of 32-kHz counter.
- ❑ Nb_hi_freq = counter_hi_freq_msb * 65536 + counter_hi_freq_lsb:
  - counter_hi_freq_msb is the MSB value of high-frequency counter.
  - counter_hi_freq_lsb is the LSB value of high-frequency counter.

Use the following procedure to gauge the 32-kHz clock versus the high-frequency clock:

1) Select gauging versus the high-frequency clock:
   - Write the gauging_ctrl[0] = 0 to select gauging versus the 12-MHz clock.

     or
   - Write the gauging_ctrl[0] = 1 to select gauging versus the external or DPLL clock.
2) Start gauging by writing the gauging_ctrl[1] = 0.
3) Wait a few seconds.
4) On reception of the gauging interrupt (low-level sensitive interrupt), stop gauging by writing the gauging_ctrl[1] = 0.
5) Check the overflow of the 32-kHz counter:

   Read it_status[2].
6) Check the overflow of the high-frequency counter:

   Read it_status[1].
7) If an overflow occurred during the process, then return to step 3 to restart the gauging and reduce the waiting time. Else continue to the next step.
8) Read the 32-kHz counter value:

   Read the counter_32_msb and counter_32_lsb registers.
9) Read the hi_freq counter value:

   Read the counter_hi_freq_msb and counter_hi_freq_lsb registers.
10) Compare the values of the counter and proceed with calibration of the 32-kHz counter.

At the end of the gauging operation, an END_GAUGING interrupt informs the MCU that gauging is stopped and values of the counters are ready to be read.

This interrupt is low-level sensitive. It is cleared on the reading of these registers.

Two other low-level-sensitive interrupts indicate whether an overflow occurred on one of the two counters during the gauging operation. One interrupt is dedicated for the 32-kHz counter, the second one for the high-frequency counter. They are also cleared on the read of the interrupt-status register.

3.6.3 Control of the 32-kHz Oscillator

The 32-kHz oscillator start-up time is configurable via the MOD_32KOSC_SW_R bits of the module-configuration-control-0 register (MOD_CONF_CTRL_0).

The 32kHz-clock source can come from either the on-chip 32-kHz oscillator or from an external 32-kHz-clock oscillator providing a clock onto the CLK32K_IN-input pin. Clock source selection depends upon the state of the CLK32K_CTRL-input pin. If this pin is driven (or tied) high, the on-chip 32-kHz oscillator is enabled as the clock source. If the pin is driven (or tied) low, then the clock must be provided externally on the CLK32K_IN pin.

The MOD_32KOSC_SW_R is a 4-bit field that sets the performance control switches of the 32-kHz oscillator (SW1, SW2, SW3, SW4). Table 4 lists the recommended control-switch settings:

Table 4. Recommended Control-Switch Settings

Oscillator Performance	**SW4**	**SW3**	**SW2**	**SW1**
Least current	1	0	0	0
Fast startup	1	0	1	1

3.6.4 Battery Failed

A battery-failed event is indicated via the BFAIL/$\overline{\text{EXT\_FIQ}}$ signal connected to the ULPD, which performs a power-down action. The ULPD processes the incoming BFAIL/$\overline{\text{EXT\_FIQ}}$ signal to an external device (via $\overline{\text{RST\_HOST\_OUT}}$) to decrease a programmable counter and to generate a shutdown signal when the counter reaches zero. It also creates an interrupt on the MPU level 1. The release of the $\overline{\text{RST\_HOST\_OUT}}$ is controlled by software. The SW_NSHUTDOWN bit in the power-control register (POWER_CTRL_REG) can be used to toggle the state of the $\overline{\text{RST\_HOST\_OUT}}$ pin to high or low level. A power-on reset condition also causes $\overline{\text{RST\_HOST\_OUT}}$ to be active low.

3.6.5 Big-Sleep and Deep-Sleep Mode

To go into deep-sleep mode, the MPU and DSP must release and unmask all interrupts that can awake the chip during this mode and they must mask all interrupts that do not awake the chip during the deep-sleep mode. Then the MPU enters the big-sleep mode. When the device reaches the idle state, it informs power management to enter into the sleep mode by setting the CHIP_IDLE signal. The state machine cuts the 12-MHz OSC1 oscillator only if the external devices do not request the oscillator clock.

3.6.6 Power-On Reset

The $\overline{\text{PWRON\_RESET}}$ signal is used to reset the entire device. This signal is resynchronized on the 32-kHz oscillator to achieve a clean reset. Then the ULPD module initiates the internal-reset sequence (a minimum of two full 32-kHz clock periods is recommended). The 32-kHz logic within the ULPD module is reset asynchronously.

3.6.7 Interrupt Wake-Up

When at least one unmask interrupt has occurred during the deep-sleep mode, a wake-up sequence is performed. In this case, the chip must leave the deep-sleep mode as soon as possible to process this interrupt. To respect the schedule of the clock enables, the following wake-up sequence is processed:

1) When the clock-management interrupt handlers detect an interrupt, the internal WAKEUP_nREQ signal is asserted.
2) The 12-MHz OSC1 oscillator is enabled and the setup counter is loaded with the setup-oscillator value.
3) When the setup counter reaches zero, the CHIP_nWAKEUP signal is used to inform the device to leave the big-sleep mode.

3.6.8 Functional Reset Generation

The ULPD generates the functional reset of the device internally from the $\overline{\text{PWRON\_RESET}}$ signal and holds it active low for a minimum of 20 12-MHz clock cycles.

3.7 32-kHz Oscillator

The 32-kHz oscillator is always on and uses a 32-kHz external quartz. The modules working with the 32-kHz clock are:

- ❑ 32-kHz timer
- ❑ Power management
- ❑ UART communication with communication processor
- ❑ PWL
- ❑ MPUIO (debouncing)/keyboard (keypad)

3.8 12- or 13-MHz Oscillator

This oscillator is to be used with a 12-MHz external quartz crystal, which is used by the clock and reset-management module and is provided to the clock reference of the DPLL1 for the MPU, DSP, TC, and peripherals. It allows the generation of the 48 MHz frequency required by the USB, camera, MMC, and UARTs. The ULPD DPLL and APLL, located in the ULPD, provide the 48 MHz frequency. By default, the USB uses the APLL. This 12-MHz clock is used as the input clock of the ULPD. The 12-MHz oscillator is on during awake and big-sleep modes. It is off during the deep-sleep mode. The power-management module handles the wake-up sequence.

When using the on-chip oscillator (normal mode), either a 12-MHz crystal or a 13-MHz crystal can be connected to the OSC1_IN and OSC1_OUT pins so that the on-chip oscillator generates a 12-MHz or a 13-MHz clock reference to the OMAP5910 device. In external-master mode, the on-chip oscillator is disabled and a 12-MHz or 13-MHz reference clock must be provided by an external oscillator connected to the OSC1_IN pin.

In both of these cases, the APLL can be configured to generate the 48-MHz clock from either a 12-MHz or a 13-MHz reference clock. If a 13-MHz reference clock is used, then use the APLL to generate the 48-MHz clock, as opposed to using the ULPD DPLL. The ULPD DPLL can only generate a 48-MHz clock when the reference is 12 MHz.

Note:13-MHz Clock

If a 13-MHz reference clock is used, then any timings detailed in this document (peripheral clocks, etc.) which are calculated based on the 12-MHz reference clock must be recalculated using the 13-MHz value.

3.9 Reset Protocol

The OMAP5910 device system reset is accomplished with a combination of hardware and software control. Individual components (or modules) can be reset by software.

There are five different sources that can cause a system reset. Three of them are internally generated, and two of them are input from the external pin. These sources are: the cold reset ($\overline{\text{PWRON\_RESET}}$ reset pin), the warm reset (from either the $\overline{\text{MPU\_RST}}$ pin or software-generated), and the watchdog reset (MPU and DSP).

- The $\overline{\text{PWRON\_RESET}}$ signal must be asserted for at least two 32-kHz-clock periods to be recognized. $\overline{\text{PWRON\_RESET}}$ indicates a power-on reset of the device. When $\overline{\text{PWRON\_RESET}}$ is asserted low, the internal power-on reset and warm reset of the MPU are active low until the 12-MHz clock (CLKIN) is on. Then the power-on reset is released after 20 CLKIN cycles, and the warm reset is released after 30 additional cycles.
- When the device is awake,the $\overline{\text{MPU\_RST}}$ controls the warm reboot of the MPU. The external-reset signal must be asserted for 30 CLKIN cycles to be recognized. The reset signal is synchronized before feeding to the reset-manager module (RSTM) that generates the internal-reset signals within the OMAP5910 device. The internal reset is asserted for at least 64 CLKIN cycles, and that clock must be running when the $\overline{\text{MPU\_RST}}$ is asserted.

Some configuration registers are only reset to their default values by certain types of resets:

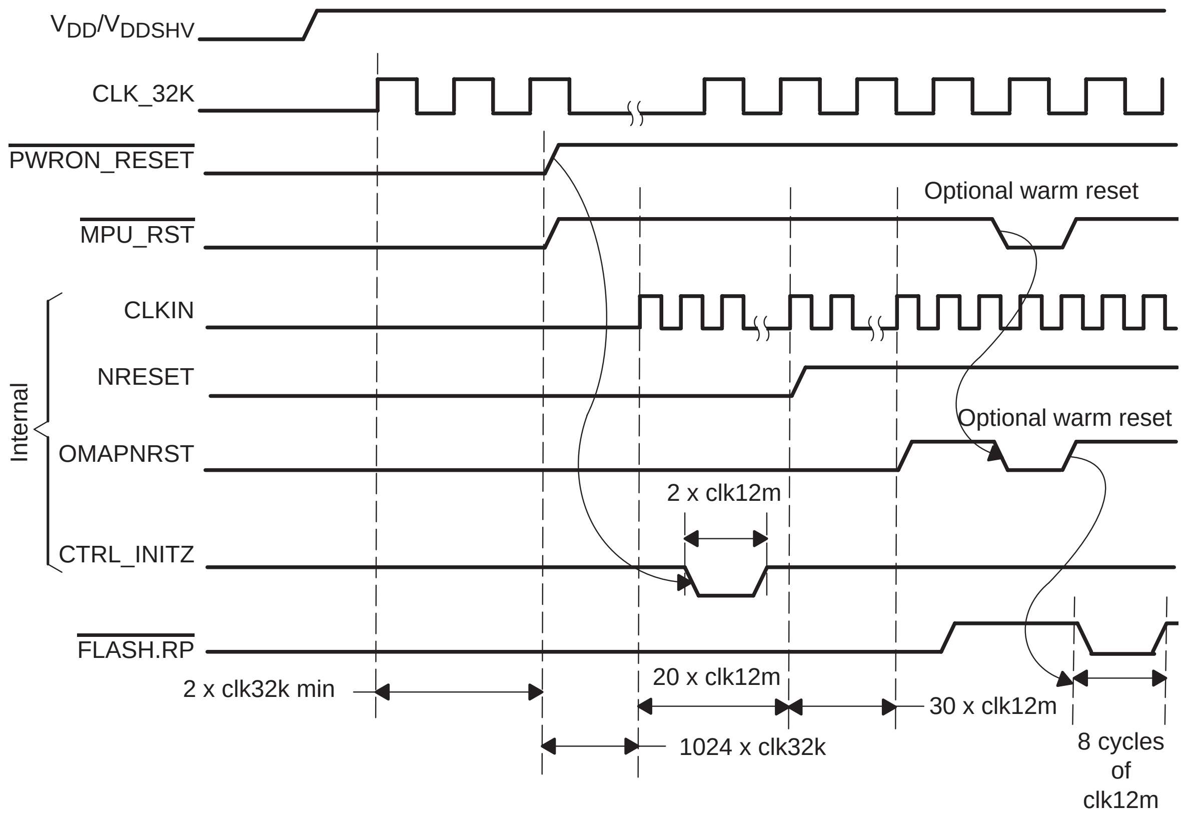
The following registers are only reset by power-on reset:

- ULPD
- Functional multiplexing configuration
- MPU level 2 interrupt handler
- MPUIO (the output register (OUTPUT_REG)
- Input/output control register (IO_CNTL))
- MCSI (the main parameters register (MAIN_PARAMETERS__REG))

DPLL registers are reset by all the resets, both hardware and software

The OMAP5910 device global-reset sequence with the behavior of the $\overline{\text{FLASH.RP}}$ pin is shown in Figure 15.

Figure 15. External Power Control During A Reset Sequence



3.9.1 Cold Reset

Cold reset is in response to the assertion of the external-reset signal ($\overline{\text{PWRON\_RESET}}$). When the reset is initiated from the pin, the OMAP5910 device is held in reset until the pin goes inactive. The reset module generates reset signals to the respective modules. All modules are put to a known state, and the RAM data is in an unknown state.

During the power-up reset, the DSP and the DSP subsystem are held in the reset mode (by hardware). The MPU boots from CS0 or CS3. Software then writes to the control registers to release the reset of the DSP subsystem once the MPU is up and running.

3.9.2 Warm Reset

A warm reset can be generated from either the $\overline{\text{MPU\_RST}}$ pin or through system software. There are several types of warm resets:

- Global: Resets the DSP, the MPU, and all internal modules through the SW_RST field, the ARM_RST field, or the DSP_EN field of the MPU-reset-control-1 register (ARM_RSTCT1).
- DSP: Resets the DSP system, with the exception of the configuration setting. The reset of the associated internal peripherals (DSP interrupt handler, timers, UART, GPIO, etc.) is controlled from the peripheral-reset signal issued by the DSP. This is done through the SW_RST field, the DSP_RST field, or the DSP_EN field of the MPU-reset-control-1 register (ARM_RSTCT1).
- MPU: Resets the MPU subsystem. The signal-reset causes an MPU reset, as does the ARM_RST field of the MPU-reset-control-1 register (ARM_RSTCT1).
- MCU: Resets the peripheral-interrupt priority-encoder registers and part of the MPUI control logic.

3.9.3 Watchdog Reset (DSP and MPU)

- MPU watchdog: A system reset (global reset) is generated when the down-counter underflows (assuming the MPU watchdog timer is configured as a watchdog timer).
- DSP watchdog: A DSP reset (reset the DSP and the DSP MMU) is generated when the down-counter underflows (assuming the DSP watchdog timer is configured as a watchdog timer)

3.9.4 Warm Reset via $\overline{\text{MPU\_RST}}$

As mentioned earlier, one type of warm reset (also called warm boot) is caused by the $\overline{\text{MPU\_RST}}$ input pin being driven active low. In a portable application initiating a warm-boot via $\overline{\text{MPU\_RST}}$ is a very convenient feature since the removal of battery-sourced power is sometimes difficult or impossible within the application. Such a warm reset is also useful in line-powered systems when a system restart is desired without cycling power.

The $\overline{\text{MPU\_RST}}$ input pin has the following characteristics:

❑ When driven active low, this pin activates a warm reset, forcing the MPU processor to reboot.

❑ The $\overline{\text{MPU\_RST}}$ input pin has a hysteresis-type input buffer.

❑ De-bouncing is not implemented

❑ The pin has a minimum pulse-width requirement (refer to appropriate data-sheet all timing requirements).

When a warm-reset condition occurs, the $\overline{\text{RST\_OUT}}$ output pin is asserted active low. The $\overline{\text{RST\_OUT}}$ output pin is always enabled (the $\overline{\text{RST\_OUT}}$ function cannot be disabled). Refer to the appropriate device datasheet for complete timing characteristics of $\overline{\text{RST\_OUT}}$ relative to $\overline{\text{MPU\_RST}}$.

The assertion of the $\overline{\text{MPU\_RST}}$ (warm reset) has the following effects:

❑ Always forces a re-boot of the MPU processor

❑ Shared GPIO logic and registers are reset.

❑ ULPD registers are reset except for the following cases:

  ■ IT_STATUS_REG: pending interrupts could possibly survive the warm-reset and become posted again after the reset condition.

  ■ DPLL register is not reset.

  ■ SETUP_ANALOG_CELL3_ULPD1_REG is not reset.

❑ ARMIO_CTL and DATA_OUT registers associated with MPUIO logic are NOT reset and retain their previous values.

❑ The appropriate bits within the ARM_SYSST register are set to indicate that the reset event was due to a warm-boot (the MPU can read ARM_SYSST to differentiate a warm-boot from a power on reset).

❑ Self-refresh of the external SDRAM is initiated if this function is enabled via the appropriate Traffic-Controller registers.

The following registers/logic are unaffected by a warm-reset condition and retain their previous state:

  ■ OMAP5910 Configuration Registers associated with device pin multiplexing

  ■ OMAP5910 Configuration Registers which enable/disable the pullup/pulldown resistors on the device pins.

- MPU Level 2 interrupt handler registers
- MPUIO control and data registers
- MCSI1 and MCSI2’s MAIN_PARAMETERS_REG registers.
- RTC (on-chip Real-time-clock) registers and logic.

Since only selective peripheral registers are reset by a warm-reset via the $\overline{\text{MPU\_RST}}$ assertion, it is recommended that the safest approach in using the $\overline{\text{MPU\_RST}}$ signal for warm-reset is to always perform a complete system re-initialization at re-boot.

The OMAP5910 device implements a single low-power pin (LOW_PWR) that indicates to the external logic that the device is in low-power mode or deep-sleep mode. The LOW_PWR signal is multiplexed on the same pin as the MPUIO5 signal, so to utilize the low-power function, user software needs to configure the pin appropriately as LOW_PWR. A warm-reset condition ($\overline{\text{MPU\_RST}}$ active) has the following effects on the LOW_PWR output pin:

Given a warm reset condition when the OMAP5910 device is awake:

- ❑ Multiplexing logic responsible for driving the LOW_PWR onto the correct pin is not reset.
- ❑ The LOW_PWR pin remains low

Given a warm-reset condition when the OMAP5910 device is in low-power or deep-sleep mode:

- ❑ Multiplexing logic responsible for driving the LOW_PWR onto the correct pin is not reset.
- ❑ The LOW_PWR pin transitions from high to low.

The OMAP5910 device implements a deep-sleep mode wherein the 12 MHz oscillator is powered down. An $\overline{\text{MPU\_RESET}}$ event will alert the ULPD module, which will turn on the 12 MHz oscillator. When this oscillator’s clock is stable, the re-boot of the MPU can begin. The time required for oscillator stability is defined by the value programmed in the analog-delay counter.

3.10 Power Control for External Devices

The idle and wake-up control module implements a power control for the external devices through the $\overline{\text{FLASH.RP}}$ output pin. The $\overline{\text{FLASH.RP}}$ signal is asserted low for eight input-clock cycles (Trl) when a global reset occurs.

Before an idle/wake-up sequence entry, the external-power control can be enabled/disabled and the time can be programmed depending on the system application. When a global-reset is asserted, the timing of the $\overline{\text{FLASH.RP}}$ signal is fixed as shown in Figure 15. The eight-cycle count starts from when the $\overline{\text{MPU\_RST}}$ pin is detected high (OMAPNRST). This means that there are actually a few extra cycles after the $\overline{\text{MPU\_RST}}$ pin is deasserted high for synchronization before the cycle count starts.

Note:

The $\overline{\text{FLASH.RP}}$ signal is low when either the $\overline{\text{PWRON\_RESET}}$ or $\overline{\text{MPU\_RST}}$ is low, and it stays low for an additional approximately 8 CLKIN cycles after the $\overline{\text{PWRON\_RESET}}$ and $\overline{\text{MPU\_RST}}$ are released.

In addition, the $\overline{\text{FLASH.RP}}$ goes low if the TC is in IDLE mode and the ARM_EWUPCT REPWR_EN field = 0.

3.11 Configuring Clocks After a Reset

After a reset, the device is in the fully-synchronous clocking mode. The DPLL1 is selected as the source for the CLK_GEN1, CLK_GEN2, and CLK_GEN3. The DPLL1 is disabled, so the device is running at the CLKIN frequency. Set the domains to operate at the desired frequencies as follows:

- Select the desired clocking mode via the CLOCK_SELECT bit of the MPU-system-status register (ARM_SYSST).
- Program each of the division modes for the DPLLs for the clock domains.
- Program the DPLLs and enable them.
- Program each domain-defined enable bit (discussed in Section 4, *Clock Generation and Reset Control Registers*). Some peripherals have additional enables for their local clocks (discussed in various peripheral chapters).

After a reset, the application software can write to the control registers via the CLOCK_SELECT[2:0] bits of the MPU-system-status register (ARM_SYSST) to switch to a desired mode of operation. However, use the system software to save the context before switching modes. For information about the switching procedure, see Section 5, *Switching Clock Modes*.

4 Clock Generation and Reset-Control Registers

The clock generation and system-reset module contains 16-bit registers for the following functions:

- ❏ Reset control
- ❏ System clocks
- ❏ Power-saving mode
- ❏ Wake-up control
- ❏ Operations
- ❏ CLKOUT pins

These registers are partitioned into the MPU (see Table 5) and DSP (see Table 17) groups.

Note:

All MPU clock generator and system-reset control registers are 32-bit accessed and all are 32-bit word aligned. All DSP control registers are 16-bit accessed by the DSP and 32-bit word aligned.

- ❏ The MPU address for these registers starts at address(hex): FFFFCE80.

Note:

All registers dedicated to the MPU are write-accessed in supervisor mode only.

- ❏ Some registers are dedicated to the DSP subsystem and can be monitored by the DSP only. Those registers are mapped to the DSP memory space starting at DSP word address (hex): 004000. They can also be accessed by the MPU through the MPUI interface.

The remaining registers are controlled by the MPU only. They are memory mapped to the MPU memory space starting at address (hex): FFFECE00.

The physical address of a register is the starting address (defined by the system) plus the offset address (given in Table 5 and following registers).

Each processor can read its associated registers at any time without affecting ongoing operations, and the registers can be written via their bits.

Table 5 lists the MPU clock/reset/power mode control registers.

Bit Width: 32

In the OMAP5910 device, the MPU is the master at all times; it has complete control of the clock generator and system reset module.

Table 5. MPU-Clock/Reset/Power Mode-Control Registers − Base Address: FFFE:CE00

Register Name	Descriptions	R/W	Size	Offset	Reset Value
ARM_CKCTL	Defines the frequency for the MPU, LCD, LCLB, MPUPER clocks	R/W	16 bits	x00	0x0000 3000
ARM_IDLECT1	Enables and defines the idle-mode entry for each clock domain	R/W	16 bits	x04	0x0000 0400
ARM_IDLECT2	Controls the clock domains individually	R/W	16 bits	x08	0x0000 0100
ARM_EWUPCT	Delay from an external device restore power with reference to the MPU clock	R/W	16 bits	x0C	0x0000 003F
ARM_RSTCT1	Initiates the S/W reset to the MPU and DSP	R/W	16 bits	x10	0x0000 0000
ARM_RSTCT2	Set the PER_EN signal	R/W	16 bits	x14	0x0000 0000
ARM_SYSST	Contains the system information such as reset status flags, processor state	R/W	16 bits	x18	0x0000 0038
ARM_CKOUT2	Reserved			0x20	

The MPU clock control register (ARM_CKCTL) defines the frequency for the ARM_CK, DSPMMU_CK, TC_CK, DSP_CK, LCD_CK, and MPUPER_CK.

Table 6. MPU-Clock-Control Register (ARM_CKCTL)

Bit	Name	Value	Description	Type	Reset Value
15	RESERVED		Reading this bit gives an undefined value, and writing to it has no effect.		
14	ARM_INTHCK_SEL		This bit controls which clock is used for the ARM_INTH_CK.	R/W	0
		0	TC_CK (this is the default and must not be changed)		
		1	Reserved		

Note: If fully-synchronous mode is selected, the divide-down bits must be programmed so that the ARMDIV, DSPDIV DSPMMUDIV, and TCDIV are all equal. At reset, these divide-down bits are all defaulted to divide by 1.

In any mode, the DSPDIV and DSPMMUDIV must be set so that the DSPMMU_CK is either equal to DSP_CK or DSP_CK/2.

In synchronous-scalable mode, the DSPMMUDIV and ARMDIV must be greater than or equal to the TCDIV.

Table 6. MPU-Clock-Control Register (ARM_CKCTL) (Continued)

Bit	Name	Value	Description	Type	Reset Value
13	EN_DSPCK		This bit allows turning on the DSP_CK while the DSP is still in a reset state.	R/W	1
		0	Disable the DSP_CK to be turned off during the reset state.		
		1	Enable the DSP_CK to be turned on during the reset state.		
12	ARM_TIMXO		Selects either the CK_GEN1 or input reference clock (CLKIN) to supply the internal MPU timers.	R/W	1
		0	The ARMTIM_CK clock frequency is the input reference clock.		
		1	The ARMTIM_CK frequency is issued from the CK_GEN1.		
11−10	DSPMMUDIV[1:0]		These read/write bits define the prescaler value from the frequency of the C*K_GEN2* to the DSPMMU clock-domain clock (DSPMMU_CK).	R/W	0
9−8	TCDIV[1:0]		These read/write bits define the prescaler value from the frequency of the CK_GEN3 to the TC clock-domain clock (TC_CK).	R/W	0
7−6	DSPDIV[1:0]		These read/write bits define the prescaler value from the frequency of the CK_GEN2 to the DSP clock-domain clock (DSP_CK).	R/W	0
5−4	ARMDIV[1:0]		These read/write bits define the prescaler value from the frequency of the CK_GEN1 to the MPU clock domain clock (ARM_CK).	R/W	0

Note: If fully-synchronous mode is selected, the divide-down bits must be programmed so that the ARMDIV, DSPDIV DSPMMUDIV, and TCDIV are all equal. At reset, these divide-down bits are all defaulted to divide by 1.

In any mode, the DSPDIV and DSPMMUDIV must be set so that the DSPMMU_CK is either equal to DSP_CK or DSP_CK/2.

In synchronous-scalable mode, the DSPMMUDIV and ARMDIV must be greater than or equal to the TCDIV.

Table 6. MPU-Clock-Control Register (ARM_CKCTL) (Continued)

Bit	Name	Value	Description	Type	Reset Value
3−2	LCDDIV[1:0]		These read/write bits define the prescaler value from the frequency of the CK_GEN3 to the LCD controller clock signal (LCD_CK).	R/W	0
1−0	PERDIV[1:0]		These read/write bits define the prescaler value from the frequency of the CK_GEN1 to the peripheral clock domain (MPUPER_CK)	R/W	0

Note: If fully-synchronous mode is selected, the divide-down bits must be programmed so that the ARMDIV, DSPDIV DSPMMUDIV, and TCDIV are all equal. At reset, these divide-down bits are all defaulted to divide by 1.

In any mode, the DSPDIV and DSPMMUDIV must be set so that the DSPMMU_CK is either equal to DSP_CK or DSP_CK/2.

In synchronous-scalable mode, the DSPMMUDIV and ARMDIV must be greater than or equal to the TCDIV.

Table 7 lists the frequency selections for the TC_CK and LCD_CK clocks.

Table 7. TC_CK and LCD_CK Frequency Selections

TCDIV[1] LCDDIV[1]	TCDIV[0] LCDDIV[0]	TC_CK Frequency LCD_CK Frequency
0	0	CK_GEN3/1
0	1	CK_GEN3/2
1	0	CK_GEN3/4
1	1	CK_GEN3/8

Table 8 lists the frequency selection for the DSP_CK clocks.

Table 8. DSP_CK Frequency Selections

DSPDIV[1]	DSPDIV[0]	DSP_CK Frequency
0	0	CK_GEN2/1
0	1	CK_GEN2/2
1	0	CK_GEN2/4
1	1	CK_GEN2/8

Table 9 lists the frequency selection for the ARM_CK and MPUPER_CK clocks.

Table 9. ARM_CK and MPUPER_CK Frequency Selections

ARMDIV[1] PERDIV[1]	ARMDIV[0] PERDIV[0]	ARM_CK Frequency MPUPER_CK Frequency
0	0	CK_GEN1/1
0	1	CK_GEN1/2
1	0	CK_GEN1/4
1	1	CK_GEN1/8

The MPU-idle-mode-entry-1 register (ARM_IDLECT1) enables and defines the idle-mode entry to each clock domain.

Table 10. MPU-Idle-Mode-Entry-1 Register (ARM_IDLECT1)

Bit	Name	Value	Description	Type	Reset Value
15−12	RESERVED		Reading these bits gives an undefined value. Writing them has no effect.		
11	SETARM_IDLE		Initiates the MPU-idle mode when written to a logical 1 and is cleared by a global reset or an interrupt request (nIRQ_SET) from the interrupt handler to the MPU processor:	R/W	1
		0	MPU is active (or in idle mode, set via the wait-for-interrupt instruction)		
		1	MPU is in idle mode		
10	WKUP_MODE		Enables the MPU to exit the idle mode from an interrupt and triggers an event on the CHIP_nWAKEUP signal:	R/W	1
		0	After an interrupt is asserted, the MPU-idle mode is exited upon a low level at the CHIP_nWAKEUP pin. The wake-up conditions wake up the OMAP5910 device out of CHIP_IDLE only if the CHIP_nWAKEUP is active.		
		1	The idle mode is exited upon an MPU interrupt (regardless of the CHIP_nWAKEUP signal). Also, any wake-up condition wakes up the OMAP5910 device out of CHIP_IDLE, regardless of the value on the CHIP_nWAKEUP signal.		

Note: When the timer/watchdog is configured as a watchdog timer, the clock is never shutdown, regardless of the value of the IDLWDT_ARM bit.

Table 10. MPU-Idle-Mode-Entry-1 Register (ARM_IDLECT1) (Continued)

Bit	Name	Value	Description	Type	Reset Value
9	IDLTIM_ARM		Selects the idle-entry mode for the internal MPU timer clock:	R/W	0
		0	Clock supplied to the timers remains active when the MPU enters the idle mode (ARM_CK stopped)		
		1	Timer clock is stopped in conjunction with the MPU clock when the idle mode is entered		
8	RESERVED		Reserved. To prevent errant behavior, always write this bit as 0.	R/W	0
7	IDLDPLL_ARM		Enables the DPLL1 to enter the idle mode when the following conditions are met: ❑ DSP is set in the global-idle mode ❑ MPU is set in the idle mode (either from a request or a wait-for-interrupt instruction) ❑ No DMA transaction is active or the TCLB_EN signal is inactive ❑ No peripheral-bus posted-write is queued ❑ Peripheral clocks are stopped	R/W	0
		0	DPLL1 remains active when the above conditions occur.		
		1	DPLL1 enters the idle mode when the above conditions are met.		
6	IDLIF_ARM		Enables the local-bus, peripheral bridge, system DMA controller, and traffic controller of the MPU subsystem to enter the idle mode whenever the MPU sets the SET_IDLE bit or executes a wait-for-interrupt instruction:	R/W	0
		0	Clocks TIPB_CK, DMA_CK, and TC_CK remain active when the MPU enters the idle mode.		
		1	Clocks TIPB_CK, DMA_CK, and TC_CK are stopped in conjunction with the MPU clock when the idle mode is entered.		

Note: When the timer/watchdog is configured as a watchdog timer, the clock is never shutdown, regardless of the value of the IDLWDT_ARM bit.

Table 10. MPU-Idle-Mode-Entry-1 Register (ARM_IDLECT1) (Continued)

Bit	Name	Value	Description	Type	Reset Value
5−3	RESERVED		Reserved. To prevent errant behavior, always write this bit as 0.	R/W	0
2	IDLPER_ARM		Selects the idle entry mode for the peripheral clock (MPUPER_CK)	R/W	0
		0	Clock remains active when the MPU enters the idle mode.		
		1	Clock is stopped in conjunction with the MPU idle mode entry.		
1	IDLXORP_ARM		Selects the idle entry mode for the 32-k or gp timer (MPU TIPB) and the peripheral clock (MPUXOR_CK):	R/W	0
		0	OS timer and MPUXOR_CK clock remain active when the MPU enters the idle mode.		
		1	OS timer and MPUXOR_CK clock are stopped in conjunction with the MPU clock when the idle mode is entered.		
0	IDLWDT_ARM		Selects the idle entry mode for the internal timer/watchdog connected to the MPU peripheral bus:	R/W	0
		0	Clock supplied to the timer/watchdog remains active when the MPU enters the idle mode.		
		1	Timer/watchdog clock is stopped in conjunction with the MPU clock when the idle mode is entered.		

Note: When the timer/watchdog is configured as a watchdog timer, the clock is never shutdown, regardless of the value of the IDLWDT_ARM bit.

The MPU-idle-mode-entry 2 register (ARM_IDLECT2) controls the clock domains independently of the MPU state.

Table 11. MPU-Idle-Mod-Entry-2 Register (ARM_IDLECT2)

Bit	Name	Value	Description	Type	Reset Value
15−11	RESERVED		Reading these bits gives an undefined values. Writing to them has no effect.		0x08
10	RESERVED		Reserved. This bit should always be written as 0.	R/W	0
9	EN_ GPIOCK		Enables the MPU-GPIO clock connected to the MPU TIPB:	R/W	0
		0	MPU-GPIO clock is stopped—the bit must be set to logic 1 to enable the clock activity		
		1	MPU-GPIO clock is active		
8	DMACK_REQ		Disables the permanently-supplied-clock to the system DMA controller to function on a clock-request basis:	R/W	1
		0	The DMA clock is shutdown when the idle mode is entered, if IDLIF_ARM = 1		
		1	The DMA clock is stopped by default (reactivated upon DMA requests only)		
7	EN_ TIMCK		Enables the MPU-timer clock connected to the MPU TIPB:	R/W	0
		0	The MPU-timer clock is stopped—the bit must be set to logic 1 to enable the clock activity		
		1	The MPU-timer clock is active and can be stopped depending on the IDLTIM_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1)		
6	EN_ APICK		Enables the MPUI clock:	R/W	0
		0	The MPUI clock is stopped—the bit must be set to logic 1 to enable the clock activity		
		1	The MPUI clock is active		
5	RESERVED		Reserved. This bit should always be written as 0.	R/W	0

Note: When the timer/watchdog is configured as a watchdog timer, the clock is never shutdown, regardless the value of the IDLWDT_ARM bit or the EN_WDTCK bit.

Table 11. MPU-Idle-Mod-Entry-2 Register (ARM_IDLECT2) (Continued)

Bit	Name	Value	Description	Type	Reset Value
4	EN_ LBCK		Enables the local bus clock:	P R/W	0
		0	The clock is stopped—it must be set to logic 1 to enable the clock activity.		
		1	The clock is active		
3	EN_ LCDCK		Enables the LCD-controller clock connected to the MPU TIPB:	P R/W	0
		0	The clock is stopped—the bit must be set to logic 1 to enable the clock activity		
		1	The clock is active		
2	EN_PERCK		Enables the peripheral clock (MPUPER_CK):	P R/W	0
		0	The clock id stopped—the bit must be set to logic 1 to authorize the clock activity		
		1	The clock is active and can be stopped depending on the IDLTIM_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1)		
1	EN_XORPCK		Enables the OS-timer clock connected to the MPU TIPB and the CLKIN-reference-peripheral clock (XORP_CK):	P R/W	0
		0	The OS-timer clock and the external XORP_CK clocks are stopped—the bit must be set to logic 1 to authorize the clock activity		
		1	The OS-timer clock and the external XORP_CK clocks are active and can be stopped depending on the IDLXORP_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1)		

Note: When the timer/watchdog is configured as a watchdog timer, the clock is never shutdown, regardless the value of the IDLWDT_ARM bit or the EN_WDTCK bit.

Table 11. MPU-Idle-Mod-Entry-2 Register (ARM_IDLECT2) (Continued)

Bit	Name	Value	Description	Type	Reset Value
0	EN_WDTCK		Enables the timer/watchdog clock connected to the MPU TIPB:	P R/W	0
		0	The clock is stopped—the bit must be set to logic 1 to authorize the clock activity		
		1	The clock supplied to the timer/watchdog is active and can be stopped depending on the IDLWDT_ARM bit of the MPU-idle-mode-entry-1 register (ARM_IDLECT1)		

Note: When the timer/watchdog is configured as a watchdog timer, the clock is never shutdown, regardless the value of the IDLWDT_ARM bit or the EN_WDTCK bit.

The MPU-external-wake-up register (ARM_EWUPCT) enables the WAKEUP signal and defines the delay from the external device to restore power with reference to the MPU clock restarting when the idle mode is exited.

Table 12. MPU-External-Wake-up Register (ARM_EWUPCT)

Bit	Name	Value	Description	Type	Reset Value
15−6	RESERVED		Reading these bits gives undefined values. Writing them has no effect.		
5	REPWR_EN		Enables the external-power-control feature:	R/W	1
		0	The $\overline{\text{FLASH.RP}}$ pin is set to logic low (Vol) when the traffic controller (TC) is in idle mode.		
		1	The $\overline{\text{FLASH.RP}}$ pin is not activated when the TC idle mode is entered		
4−0	EXTPW[4:0]		Define the delay from the $\overline{\text{PWRON\_RESET}}$ pin going high to the clock's restarting: The delay is calculated as follows: t_W (Wake-up time) = [EXTPWR(field value) ± 1] x *CLKIN* (period) with EXTPWR = 0 to 31	R/W	1

The MPU-reset-control-1 register (ARM_RSTCT1) initiates the software reset to the DSP and to the MPU.

Table 13. MPU-Reset-Control-1 Register (ARM_RSTCT1)

Bit	Name	Value	Description	Type	Reset Value
15−4	RESERVED		Reading these bits gives undefined values. Writing to them has no effect.		
3	SW_RST		Resets the DSP, MPU, and peripherals (the bit is always read 0):	R/W	0
		0	The DSP, MPU, and peripheral clock domain is enabled		
		1	The DSP, MPU, and peripherals are reset—once set to logic 1 by the MPU processor, this bit returns to logic 0 once the reset completes.		
2	DSP_RST		Resets the priority registers (TIPB module), EMIF configuration registers, and the MPUI control logic (partially) in the DSP. This bit is set by the external reset pins and is released by writing a logic 1 in the register (use for MPUI boot).	R/W	0
		0	Priority, EMIF configuration registers, and the MPUI are reset.		
		1	Priority and EMIF configuration registers can be programmed.		
1	DSP_EN		Resets the DSP:	R/W	0
		0	Resets the DSP, excluding configuration setting, and maintains the reset state as long as this bit is asserted low		
		1	Enables the DSP—after a global reset, the bit must be set to a logical 1 to enable the DSP.		
0	ARM_RST		Resets the MPU (bit is always read 0):	R/W	0
		0	MPU clock domain is enabled		
		1	MPU is reset—once set to logic 1 by the MPU processor, the bit returns to logic 0 on the next cycles.		

Note: Writing the DSP_EN bit to 0 and the ARM_RST bit to 1 together initiate a global-software reset.

The MPU-reset-control-2 register (ARM_RSTCT2) sets the PER_EN signal that resets the peripherals attached to the MPU.

Table 14. MPU-Reset-Control-2 Register (ARM_RSTCT2)

Bit	Name	Value	Description	Type	Reset Value
15−1	RESERVED		Reading these bits gives undefined values. Writing to them has no effect.		
0	PER_EN		Controls the MPUPER_nRST signal used to reset and/or enable the peripherals connected to the MPU TIPB:	R/W	0
		0	The ARMPER_nRST signal is active		
		1	The ARMPER_nRST signal is inactive		

The MPU-system-status register (ARM_SYSST) contains the system information such as the processor state, chip configuration, and reset-status flags.

Table 15. MPU-System-Status-Register (ARM_SYSST)

Bit	Name	Value	Description	Type	Reset Value
15−14	RESERVED		Reading these bits gives undefined values. Writing to them has no effect.		
13−11	CLOCK_SELECT [2−0]		The CLOCK_SELECT bits indicate the current clocking-scheme selection: the application can switch the OMAP5910 clocking scheme by writing to these bits. These bits are at logic 0 after reset (select fully-synchronous clocking scheme) (see Table 16).	R/W	0
10−7	RESERVED				
6	IDLE_DSP		Indicates the DSP state:	R	0
		0	The DSP is active		
		1	The DSP is in the global-idle state		
5	POR		Indicates (in conjunction with the EXT_RST bit) whether or not a power-on reset (cold start) has occurred. Writing it to logic 0 clears this bit. This bit cannot be written to logic 1 from the TIPB interface:	R/C	0
		0	No power-on reset is detected		
		1	A power-on reset occurred		

Table 15. MPU-System-Status-Register (ARM_SYSST) (Continued)

Bit	Name	Value	Description	Type	Reset Value
4	EXT_RST		Indicates that an external reset has been asserted. Writing it to logic 0 clears this bit. This bit cannot be written to logic 1 from the TIPB:	R/C	0
		0	No external reset is detected		
		1	An external reset is asserted		
3	ARM_MCRST		Indicates whether or not an MPU reset has occurred. This bit is cleared to 0 upon a reset pulse asserted at the CHIP_nRESET signal, or by writing to it a logic 0. This bit cannot be written to logic 1 from the TIPB interface:	R/C	0
		0	The MPU has not been reset.		
		1	The MPU has been reset.		
2	ARM_WDRST		Indicates whether or not a reset has been asserted due to an MPU timer/watchdog underflow. This bit is cleared to logic 0 upon a reset pulse asserted at the CHIP_nRESET signal, or by writing it to logic 0. This bit cannot be written to logic 1 from the peripheral-bus interface:	R/C	0
		0	An MPU-timer/watchdog underflow has not occurred.		
		1	An MPU-timer/watchdog underflow has generated a reset.		
1	GLOB_SWRST		Indicates whether or not a reset has been asserted due to a global-software reset (DSP_EN and ARM_RST are set together). This bit is cleared to logic 0 upon an external-reset pulse asserting at the CHIP_nRESET signal, or by writing it to logic 0. This bit cannot be written to logic 1 from the peripheral-bus interface:	R/C	0
		0	A global-software reset has not been requested.		
		1	A global-software reset has been requested.		

Table 15. MPU-System-Status-Register (ARM_SYSST) (Continued)

Bit	Name	Value	Description	Type	Reset Value
0	DSP_WDRST		Indicates whether or not a reset has been asserted due to a DSP-timer/watchdog underflow. This bit is cleared to logic 0 upon a reset pulse asserting at the CHIP_nRESET signal, or by writing it to logic 0. This bit cannot be written to logic 1 from the peripheral-bus interface:	R/C	0
		0	A DSP-timer/watchdog underflow has not occurred.		
		1	A DSP-timer/watchdog underflow has generated a reset.		

Table 16 lists the clocking schemes for the MPU-system-status register (ARM_SYSST).

Table 16. Clocking Schemes for OMAP5910

CLOCK_SELECT[2]	CLOCK_SELECT[1]	CLOCK_SELECT[0]	CLOCK SCHEME
0	0	0	Fully synchronous
0	0	1	Reserved
0	1	0	Synchronous scalable
0	1	1	Reserved
1	x	x	Reserved

DSP Base word address: 0x4000 − Bit Width: 16

Table 17. DSP-Clock/Reset/Power Mode-Control Registers

Register Name	Descriptions	R/W	Size	DSP Address	MPU Address	Reset Value
DSP_IDLECT1	DSP idle select 1	R/W	16 bits	00:4002	E100:8004	0x0000
DSP _IDLECT2	DSP idle select 2	R/W	16 bits	00:4004	E100:8008	0x0000
Reserved				00:4006		0x0000
Reserved				00:4008		0x0000
DSP _RSTCT2	DSP reset control	R/W	16 bits	00:400A	E100:8014	0x0000
DSP _SYSST	DSP system status	R/W	16 bits	00:400C	E100:8018	0x0000

Table 17. DSP-Clock/Reset/Power Mode-Control Registers (Continued)

Register Name	Descriptions	R/W	Size	DSP Address	MPU Address	Reset Value
Reserved				00:400E		0x0000
Reserved				00:4010		0x0000

The DSP-domain-peripheral-clock setup and the external-module resetfunctions are controlled by the DSP through these registers.

The DSP-clock-control-register (DSP_CKCTL) defines the frequency selection for the DSP_GPIO_CK and the DSPTIM_CK.

Table 18. DSP-Clock -Control Register (DSP_CKCTL) — Offset Address: 0x00

Bit	Name	Value	Description	Type	Reset Value
15−9	RESERVED		Reading these bits gives an undefined value. Writing to them has no effect.		
8	TIMXO		Selects either a CK_GEN2 frequency clock or an input reference clock (CLKIN) to supply the DSP-timer peripherals.	R/W	1
		0	The DSPTIM_CK-clock frequency is the input reference clock.		
		1	The DSPTIM_CK frequency is issued from the CK_GEN2/2.		
7	GPIOXO		Selects either a subfrequency issued from the CK_GEN2 or the input reference clock (CLKIN) to supply the GPIO peripheral.	R/W	1
		0	The DSP_GPIO_CK-clock frequency is the input reference clock.		
		1	The DSP_GPIO_CK frequency is issued from the CK_GEN2 and defined by the GPIODIV field value.		
6−5	GPIODIV[1:0]		These read/write bits define the prescaler value from the CK_GEN2 to the GPIO-clock signal (GPIO_CK).	R/W	0
4−0	RESERVED		Reserved. These bits should always be written as 0.		

Table 19 lists the selection for the DSP_GPIO_CK (GPIOXO = 1).

Table 19. GPIO_CK Selections

GPIODIV(1)	GPIODIV(0)	DSP_GPIO_CK Frequency
0	0	CK_GEN2/1
0	1	CK_GEN2/2
1	0	CK_GEN2/4
1	1	CK_GEN2/8

The DSP-idle-mode-entry-1 register (DSP_IDLECT1) enables and defines the idle-mode entry/exit to each clock domain.

Table 20. DSP-Idle-Mod-Entry-1 Register (DSP_IDLECT1) − Offset Address: 0x04

Bit	Name	Value	Description	Type	Reset Value
15−9	RESERVED		Reading these bits gives undefined values. Writing to them has no effect.		
8	IDLTIM_DSP		Selects the idle-entry mode for the internal DS-timer clock (DSPTIM_CK).	R/W	0
		0	The DSPTIM_CK clock remains active when the DSP enters the idle mode.		
		1	The DSPTIM_CK clock is stopped in conjunction with the DSP clock when the idle mode is set (DSP_IDLE signal is asserted high).		
7	RESERVED			R/W	0
6	RESERVED		Reserved. To prevent errant behavior, this bit should always be written as 1.	R/W	1
5−2	RESERVED		Reserved. To prevent errant behavior, these bits should always be written as o.	R/W	0
1	IDLXORP_DSP		Selects the idle entry mode for the reference-peripheral clock (DSPXOR_CK).	R/W	0
		0	The DSPXOR_CK clock remains active when the DSP enters the idle mode.		
		1	The DSPXOR_CK clock is stopped in conjunction with the DSP clock when the idle mode is entered.		

Note: When the timer/watchdog is configured as a watchdog timer, the clock is never shutdown, regardless of the value of the IDLWDT_DSP bit.

Table 20. DSP-Idle-Mod-Entry-1 Register (DSP_IDLECT1) − Offset Address: 0x04

Bit	Name	Value	Description	Type	Reset Value
0	IDLWDT_DSP		Selects the idle-entry mode for the timer/watchdog connected to the DSP TIPB.	R/W	0
		0	The clock supplied to the timer/watchdog remains active when the DSP enters the idle mode.		
		1	The timer/watchdog clock is stopped in conjunction with the DSP clock when the idle mode is entered.		

Note: When the timer/watchdog is configured as a watchdog timer, the clock is never shutdown, regardless of the value of the IDLWDT_DSP bit.

The DSP-idle-mode-entry-2 register (DSP_IDLECT2) disables the clock domains individually and independently of the DSP state.

Table 21. DSP-Idle-Mode-Entry-2 Register (DSP_IDLECT2) − Offset Address: 0x08

Bit	Name	Value	Description	Type	Reset Value
15−6	RESERVED		Reading these bits gives undefined values. Writing to them has no effect.		
5	EN_TIMCK		Enables the DSP-timer clock (DSPTIM_CK)	R/W	0
		0	The DSPTIM_CK clock is stopped. This bit must be set to logic 1 to enable the clock activity.		
		1	The DSPTIM_CK clock is active and can be stopped, depending on the DSP_IDLECT1 IDLTIM_DSP bit.		
4	EN_GPIOCK		Enables the GPIO-peripheral clock (GPIO_CK).	R/W	0
		0	The GPIO_CK clock is stopped. The bit must be set to logic 1 to enable the clock activity.		
		1	The GPIO_CK clock is active. The GPIO_CK clock must be disabled by setting the EN_GPIOCK = 0 before sleep modes can be entered.		
3−2	RESERVED		Reserved. These bits should always be written as 0.		

Table 21. DSP-Idle-Mode-Entry-2 Register (DSP_IDLECT2) − Offset Address: 0x08 (Continued)

Bit	Name	Value	Description	Type	Reset Value
1	EN_XORPCK		Enables the DSPXOR_CK-reference clock.	R/W	0
		0	The DSPXOR_CK clock is stopped. The bit must be set to logic 1 to enable the clock activity.		
		1	The DSPXOR_CK clock is active and can be stopped, depending on the DSP_IDLECT1 IDLXORP_DSP bit.		
0	EN_WDTCK		Enables the DSPWDT_CK-reference clock.	R/W	0
		0	The DSPWDT_CK clock is stopped. The bit must be set to logic 1 to enable the clock activity.		
		1	The DSPWDT_CK clock is active and can be stopped, depending on the DSP_IDLECT1 IDLWDT_DSP bit.		

The DSP-reset-control-2 register (DSP_RSTCT2) sets the PER_EN signal that resets the peripherals attached to the DSP.

Table 22. DSP-Reset-Control-2 Register (DSP_RSTCT2) − Offset Address: 0x14

Bit	Name	Value	Description	Type	Reset Value
15−1	RESERVED		Reading these bits gives undefined values. Writing to them has no effect.		
0	PER_EN		This read/write bit controls the DSPPER_nRST signal that can be used to reset and/or enable the peripherals connected to the DSP.	R/W	0
		0	Writing a logic 0 sets the DSPPER_nRST signal to active.		
		1	Writing a logic 1 sets the DSPPER_nRST signal to inactive.		

The DSP-system-status register (DSP_SYSST) contains the system information.

Table 23. DSP-System-Status Register (DSP_SYSST) − Offset Address: 0x18

Bit	Name	Value	Description	Type	Reset Value
15−14	RESERVED		Reading these bits gives undefined values. Writing to them has no effect.		
13−11	CLOCK_SELECT (2−0)		These read-only bits reflect the CLOCK_SELECT and indicate the current clocking-scheme selection.	R	0
10−7	RESERVED				
6	IDLE_ARM		This read-only bit indicates the MPU state.	R	0
		0	The MPU is active		
		1	The MPU is in idle state		
5	POR		This read/clear-only status bit indicates (in conjunction with the EXT_RST bit) whether or not a power-on reset (cold start) has occurred. Writing it to logic 0 clears this bit. This bit cannot be written to logic 1 from the peripheral-bus interface.	R/C	0
		0	No power-on reset has been detected.		
		1	A power-on reset has occurred.		
4	EXT_RST		This read/clear-only status bit indicates whether or not an external reset has been asserted. Writing it to logic 0 clears this bit. This bit cannot be written to logic 1 from the TIPB interface.	R/C	0
		0	No external reset detected.		
		1	An external reset has been asserted.		
3	DSP_ARM_RST		This read/write bit is used by the DSP to hold the MPU in reset.	R/C	0
		0	The MPU is enabled. This is the default value after reset.		
		1	Reset the MPU		

Notes: 1) This bit is only to be used for testing/debugging purposes only.

2) In the OMAP5910 device, the DSP_EN and ARM_RST bits (located in the ARM_RSTCT1) must be set together to activate the global-software reset. Setting the SW_RST bit only (DSP_RSTCT1) results in a global-software reset flag.

Table 23. DSP-System-Status Register (DSP_SYSST) − Offset Address: 0x18 (Continued)

Bit	Name	Value	Description	Type	Reset Value
2	ARM_WDRST		This read/clear-only status bit indicates whether or not a reset has been asserted due to a the MPU timer/watchdog underflow. This bit is cleared to logic 0 upon an external reset pulse asserting at the CHIP_nRESET signal, or by writing it to logic 0. This bit cannot be written to logic 1 from the peripheral bus interface.	R/C	0
		0	MPU timer/watchdog underflow has not occurred.		
		1	MPU timer/watchdog underflow has generated a reset.		
1	GLOB_SWRST		This read/clear-only status bit indicates whether or not a reset has been asserted due to a global-software reset. This bit is cleared to logic 0 upon an external reset pulse asserting at the CHIP_nRESET signal, or by writing it to logic 0. This bit cannot be written to logic 1 from the TIPB interface. See Note 2.	R/C	0
		0	A global-software reset has not been requested.		
		1	A global-software reset has been requested.		
0	DSP_WDRST		This read/clear-only status bit indicates whether or not a reset has been asserted due to a DSP timer/watchdog underflow. This bit is cleared to logic 0 upon an external reset pulse at the CHIP_nRESET signal or by writing it to logic 0. This bit cannot be written to logic 1 from the peripheral bus interface.	R/C	0
		0	DSP timer/watchdog underflow has not occurred.		
		1	DSP timer/watchdog underflow has generated a reset.		

Notes: 1) This bit is only to be used for testing/debugging purposes only.

2) In the OMAP5910 device, the DSP_EN and ARM_RST bits (located in the ARM_RSTCT1) must be set together to activate the global-software reset. Setting the SW_RST bit only (DSP_RSTCT1) results in a global-software reset flag.

4.1 DPLL Operation Mode Registers

The digital phase-locked loop (DPLL) can be operated either in bypass mode, in lock mode, or in idle mode.

- In lock mode, the DPLL synthesizes a frequency clock (CLKOUT) from a fixed reference-clock signal (CLKREF). The output frequency is an integer multiple or fractional multiple (m/n, respectively PLL_MULT and PLL_DIV-bit field) of the input reference. With 1 < m < 31 and 1 < n < 4, the frequency-output ranges from CLKREF /4 to 31x CLKREF.
- In bypass mode, the DPLL-output clock in bypass mode can be the CLKREF (input clock), the CLKREF/2, or the CLKREF/4 depending on the BYPASS_DIV-bit-field value.
- In idle mode, the DPLL circuitry is disabled. The output clock is held in a high static level and the configuration data is maintained.

The mode (bypass or lock) and the frequency selections are defined via the memory-mapped control-register bits. Programming is performed through the TIPB and is initiated in the MPU.

The idle mode is entered upon an asynchronous-idle-signal request (idle).

Note:

The DPLL-idle-mode-entry/exit timing is dependent upon the input/output frequency ratio selection.

The input-reference-clock signal must be active for (at least) 24 input-clock cycles from the idle request (idle-rising edge) before the DPLL-idle setup has completed (idle-ack high).

Once in the idle mode, the clock source can be stopped (either in a high or a low level), but the reference-clock source must be restarted before releasing the idle mode.

When the idle mode is exited, the DPLL is set in bypass mode and the output clock signal is valid after a maximum of 10 input reference-clock cycles. If the DPLL was synthesizing a frequency prior to entering the idle state, then the DPLL switches from the bypass mode (frequency set /BYPASS_DIV) to the synthesizer mode (frequency set/PLL_MULT and PLL_DIV) when the lock state is reacquired.

4.1.1 To Switch From DPLL Bypass to DPLL Lock Mode

For proper operation in lock mode, use the following steps:

Step 1: Set the desired Multiplier bits (11:7) and Divider bits (6:5) in the DPLL Control Register.

Step 2: Set (write 1) in the PLL_ENABLE bit (bit 4 in the DPLL Control Register).

Step 3: Poll the lock bit (bit 0) until 1 is read by the MPU.

Step 4: software may have a timeout counter to break the polling loop in case the DPLL does not lock.

4.1.2 To Switch From DPLL Lock Mode to DPLL Bypass Mode

For proper operation in bypass mode, use the following steps:

Step 1: Write 0 to the PLL_ENABLE bit (bit 4 in the DPLL Control Register2).

Step 2: Poll the lock bit (bit 0) until the 0 is read by the MPU.

Step 3: Software may have a timeout counter to break the polling loop in case the DPLL does switch to bypass mode.

Table 24 lists the DPLL-control registers. Table 25 describes the register bits.

The MPU base addresses for the DPLL-control registers are:

DPLL1:	0xFFFE:CF00
Bit width:	16 bits

Writing to the control register (CTL_REG) causes the DPLL to immediately switch to the bypass mode, if not in the idle state. If the PLL_ENABLE bit of the control register is set, it begins its sequence to enter the locked mode. This prevents being able to change the multiply or divide values without reentering the DPLL lock sequence.

Table 24. DPLL-Control Registers

Register Name	Descriptions	R/W	Size	Offset	Reset Value
CTL_REG	DPLL-control register	R/W	16 bits	x00	0x2002

Table 25. DPLL-Control Register (CTL_REG)

Bit	Name	Description	Type	Reset Value
15−14	RESERVED	This bit is reserved and is set to 0.		
13	IOB	Initialize on break. When high, the DPLL switches to the bypass mode and starts a new locking sequence if the DPLL core indicates that it lost lock. When set low, the DPLL continues to output the synthesized clock even if the core indicates it has lost lock, but the BREAKLN is active low. The power-on value is 1.		
12	RESERVED	This bit is reserved and set to 0.		
11−7	PLL_MULT[4−0]	The DPLL multiply value. The maximum clock-out frequency is 31 * CLKREF.		
6−5	PLL_DIV[1:0]	The DPLL divide value. PLL_DIV[1:0] = 00: CLKOUT=CLKREF 01: CLKOUT=CLKREF/2 10: CLKOUT = CLKREF/3 11: CLKOUT = CLKREF/4 The minimum CLKOUT frequency is 0.25 * CLKREF. When the PLL_MULT[4:0] is equal to 0 or 1, the CLKOUT is not synthesized by DPLL but by simply a divided down version of the CLKREF. Only lock mode is affected.		
4	PLL_ENABLE	Setting the PLL_ENABLE bit to 1 requests the DPLL to enter the LOCK mode. It enters the LOCK mode only after it has synthesized the desired frequency. Clearing the bit to 0 causes the DPLL to switch back to the bypass mode.		
3−2	BYPASS_DIV[1:0]	Determines the clkout frequency when in the BYPASS mode. BYPASS_DIV[1:0] = 00: CLKOUT = CLKREF 01: CLKOUT = CLKREF/2 1X: CLKOUT = CLKREF/4		

Table 25. DPLL-Control Register (CTL_REG)

Bit	Name	Description	Type	Reset Value
1	BREAKLN	When the BREAKLN = 0, the DPLL has broken the lock condition for some unknown reason. If and when the lock condition is restored or a write-to-control register occurs, the BREAKLN returns to 1.		
0	LOCK	When the LOCK = 1, the DPLL is in lock mode and the clkout is the desired synthesized-clock frequency. When the LOCK = 0, the DPLL is in bypass mode and the clkout contains a divided-down output clock. If the PLL_MULT = 0 or 1, the oscillators in the DPLL-core are not activated and the duty cycle is not ensured.		

Table 26 lists the ULPD registers. Table 27 through Table 41 describe the register bits.

Bit Width: 32

Table 26. ULPD Registers − MPU Base Address: FFFE:0800

Name	Descriptions	R/W	Size	Offset	Reset Value
COUNTER_32_LSB	Lower value of number of ticks from the 32-kHz clock	R	16 bits	x00	0x0001
COUNTER_32_MSB	Upper value of number of ticks from the 32-kHz clock	R	16 bits	x04	0x0000
COUNTER_HIGH_FREQ_LSB	Lower value of number of ticks from the high-frequency clock	R	16 bits	x08	0x0001
COUNTER_ HIGH_FREQ _MSB	Upper value of number of ticks from the high-frequency clock	R	16 bits	x0C	0x0000
GAUGING_CTRL_REG	Drives the gauging functionality	R/W	16 bits	0x10	0x0000
IT_STATUS_REG	Interrupt-status register	R	16 bits	0x14	0x0000
Reserved			8 bits	0x18	0x01
Reserved			8 bits	0x1C	0x01
Reserved			8 bits	0x20	0x01
SETUP_ANALOG_CELL3_ULPD1_REG	Number of 32-kHz clocks to wake up	R/W	16 bits	0x24	0x03FF

Table 26. ULPD Registers − MPU Base Address: FFFE:0800 (Continued)

Name	Descriptions	R/W	Size	Offset	Reset Value
Reserved			8 bits	0x2C	0x01
Reserved			8 bits	0x28	0x01
CLOCK_CTRL_REG	Manages the clock output and inactive values	R/W	16 bits	0x30	0x0000
SOFT_REQ_REG	Manages the software clock requests	R/W	16 bits	X34	0x0000
COUNTER_32_FIQ_REG	Number of 32-kHz clocks to delay the active the modem shut-down signal after receiving an active EXT_FIQ signal	R/W	16 bits	X38	0x0001
DPLL_CTRL_REG	48-MHz DPLL	R/W	16 bits	X3C	0x2211
STATUS_REQ_REG	Status of the hardware requests	R	16 bits	0x40	U
LOCK_TIME_REGISTER	Defines the lock time when the APLL is selected	R/W	16 bits	0x48	0x960
APLL_CTRL_REG	This register allows switching between the APLL and DPLL, and controls all input of the APLL.	R/W	16 bits	0x4C	U
POWER_CTRL_REG	Power-control register	R/W	16 bits	0x50	0x8

The counter-32-LSB register (COUNTER_32_LSB_REG) represents the lower value of the number of ticks from the 32-kHz clock during the gauging time.

Table 27. Counter-32-LSB Register (COUNTER_32_LSB_REG)

Bit	Name	Type	Reset Value
15−0	COUNTER_32_LSB	R	0x0001

The counter-32-MSB register (COUNTER_32_MSB_REG) represents the upper value of the number of ticks from the 32-kHz clock during the gauging time.

Table 28. Counter-32-MSB Register (COUNTER_32_MSB_REG)

Bit	Name	Type	Reset Value
15−0	COUNTER_32_LSB	R	0x0000

The counter-high-frequency-LSB register (COUNTER_HIGH_FREQ_LSB_REG) represents the lower value of the number of ticks of the high-frequency clock during the gauging time.

Table 29. Counter-High-Frequency-LSB Register (COUNTER_HIGH_FREQ_LSB_REG)

Bit	Name	Type	Reset Value
15−0	COUNTER_HIGH_FREQ_LSB	R	0x0001

The counter-high-frequency-MSB register (COUNTER_HIGH_FREQ_MSB_REG) represents the upper value of the number of ticks from the high-frequency clock during the gauging time.

Table 30. Counter-High-Frequency-MSB Register (COUNTER_HIGH_FREQ_MSB_REG)

Bit	Name	Type	Reset Value
15−0	COUNTER_HIGH_FREQ_LSB	R	0x0000

The gauging-control register (GAUGING_CTRL_REG) controls the gauging activity. It start/stops and selects the clock used as the high-frequency clock.

Table 31. Gauging-Control Register (GAUGING_CTRL_REG)

Bit	Name	Value	Description	Type	Reset Value
1	SELECT_HI_ FREQ_CLOCK	0	Use the 12-MHz clock for the high-frequency clock	R/W	0
		1	Reserved. Do not use this setting.		
0	GAUGING_EN	0	Stops the gauging activity	R/W	0
		1	Enables the gauging activity		

The setup-analog-cell3-ULPD1 register (SETUP_ANALOG_CELL3_ULPD1_REG) provides the number of 32-kHz clock periods until the ULPD wakes up the device when a wake-up request is made. See Figure 14.

Table 32. Setup-Analog-Cell-ULPD1 Register (SETUP_ANALOG_CELL3_ULPD1_REG)

Bit	Name	Type	Reset Value
15−0	SETUP_ANALOG_CELL3	R/W	0x3FF

The interrupt-status register (IT_STATUS_REG) provides the status and the source of the ULPD interrupts.

Table 33. Interrupt-Status Register (IT_STATUS_REG)

Bit	Name	Description	Type	Reset Value
3	IT_WAKEUP_USB	The wake-up interrupt from the USB function is shared with the ULPD-gauging interrupt (MPU level 2 interrupt IRQ_24).	R	0x0
2	OVERFLOW_32	An overflow occurred on the 32-kHz counter during gauging.	R	0x0
1	OVERFLOW_HI_FREQ	An overflow occurred in the high-frequency counter during gauging versus the high-frequency clock.	R	0x0
0	IT_GAUGING	End of the gauging interrupt. Informs the MPU that gauging is stopped and that it can read the value of the high and low frequency counters.	R	0x0

The clock-control register (CLOCK_CTRL_REG) manages the clock output and inactive values.

Table 34. Clock-Control Register (CLOCK_CTRL_REG)

Bit	Name	Value	Description	Type	Reset Value
5	DIS_USB_PVCI_CLK	0	Enables the USB function clock for the FAC counter	R/W	0x0
		1	Disables the USB function clock for the FAC counter		
4	USB_MCLK_EN	0	Disables the USB.CLKO	R/W	0
		1	Enables the USB.CLO		
3	RESERVED		Reserved. This bit should always be written as 0.	R/W	0x0
2	SDW_MCLK_INV	0	The BCLK is low when inactive.	R/W	0
		1	The BCLK is high when inactive.		

Table 34. Clock-Control Register (CLOCK_CTRL_REG) (Continued)

Bit	Name	Value	Description	Type	Reset Value
1	COM_MCLK_ INV	0	The MCLK is low when inactive.	R/W	0
		1	The MCLK is high when inactive		
0	MODEM_32K_ EN	0	Disables the 32-kHz frequency on the UART clock	R/W	0
		1	Enables the 32-kHz frequency on the UART clock		

The software-clock-request register (SOFT_REQ_REG) manages software clock requests.

Table 35. Software-Clock-Request Register (SOFT_REQ_REG)

Bit	Name	Value	Description	Type	Reset Value
4	USB_REQ_EN	0	Disables the USB function hardware DPLL request	R/W	0x1
		1	Enables the USB function hardware DPLL request		
3	SOFT_USB_REQ	0	No software request for clocking on the USB.CLK0	R/W	0
		1	Software request for clocking on the USB.CLK0		
2	SOFT_SDW_REQ	0	No software request for clocking on the BCLK	R/W	0
		1	Software request for clocking on the BCLK		
1	SOFT_COM_REQ	0	No software request for clocking on the MCLK	R/W	0
		1	Software request for clocking on the MCLK		
0	SOFT_DPLL_REQ	0	Software request for clocking on the 48-MHz DPLL	R/W	0
		1	Software request for clocking on the 48-MHz DPLL (except no software request is possible when the PLL_ENABLE = 0 in the DPLL_CTRL_REG)		

The counter-32-FIQ register (COUNTER_32_FIQ_REG) represents the number of 32-kHz clocks to delay before activating the $\overline{\text{RST\_HOST\_OUT}}$ after receiving an active BFAIL/$\overline{\text{EXT\_FIQ}}$ signal.

Table 36. Counter-32-FIQ Register (COUNTER_32_FIQ_REG)

Bit	Name	Type	Reset Value
7−0	COUNTER_32_FIQ	R/W	0x01

This is the control register for the 48-MHz DPLL.

The reset multiply and divide settings are fixed for x4 operation to achieve a 48-mHz clock. The DPLL must always remain enabled when it is used to generate the 48-mHz clock for the USB, UARTs, or other peripherals

Writing to the DPLL-control register (DPLL_CTRL_REG) causes the DPLL to switch immediately to the bypass mode, if not in idle state. If the PLL_ENABLE bit of the control register is set, it begins its sequence to enter the locked mode. This prevents being able to change the multiply or divide values without reentering the DPLL lock sequence.

Table 37. DPLL-Control Register (DPLL_CTRL_REG)

Bit	Name	Description	Type	Reset Value
15−14	RESERVED	This bit is reserved and set to 0.		
13	IOB	Initialize on break. When high, the DPLL switches to bypass mode and starts a new locking sequence if the DPLL core ever indicates that it lost the locked condition. When set low, the DPLL continues to output the synthesized clock, even if the core indicates it has lost the locked condition, but the BREAKLN is active low. The power-on value is 1.	R/W	1
12	RESERVED	This bit is reserved and set to 0.		
11−7	PLL_MULT[4:0]	This bit is reserved and always written to its reset value.	R/W	0x0
6−5	PLL_DIV[1:0]	This bit is reserved and always written to its reset value.	R/W	0x0
4	PLL_ENABLE	Setting the PLL_ENABLE bit to 1 requests the DPLL to enter the LOCK mode. It enters the lock mode only after it has synthesized the desired frequency. Clearing the bit to 0 causes the DPLL to switch back to the bypass mode.	R/W	1
3−2	BYPASS_DIV[1:0]	This bit is reserved and always written to its reset value.	R/W	0x0
1	BREAKLN	When the BREAKLN = 0 , the DPLL has broken the locked condition for some unknown reason. If and when the locked condition is restored or a write-to-control register occurs, the BREAKLN returns to 1.	R	0x1
0	LOCK	When the LOCK = 1, the DPLL is in the locked mode and the clkout is the desired synthesized clock frequency. When the LOCK = 0, the DPLL is in the bypass mode and the clkout contains a divided-down output clock.	R	0x0

The status-request register (STATUS_REQ_REG) indicates the status of the hardware requests. The reset value of these registers depends on what requests are being made.

Table 38. Status-Request Register (STATUS_REQ_REG)

Bit	Name	Value	Description	Type	Reset Value
13	MODEM_ NSHUTDOWN		Status of the $\overline{\text{RST\_HOST\_OUT}}$ output pin	R	
12	MMC_DPLL_REQ		The DPLL wake-up request from the MMC	R	
11	UART3_DPLL_REQ		The DPLL wake-up request for the UART3	R	
10	UART2_DPLL_REQ		The DPLL wake-up request for the UART2	R	
9	UART1_DPLL_REQ		The DPLL wake-up request for the UART1	R	
8	USB_HOST_ DPLL_REQ		The 12-MHz clock for the DPLL wake-up requested by the USB host	R	
7	CAM_DPLL_ MCLK_REQ	0	No request for the 48-MHz DPLL wake-up by the camera interface	R	
		1	Indicates a request for the 48-MHz DPLL wake-up by the camera interface		
6	USB_DPLL_ MCLK_REQ	0	No request for the 48-MHz DPLL wake-up by the USB interface	R	
		1	Indicates a request for the 48-MHz DPLL wake-up by the USB interface		
5	USB_ MCLK_REQ	0	No clock request by the USB host	R	
		1	Indicates a clock request by the USB host		
4	SDW_ MCLK_REQ	0	No clock request by the BCLKREQ	R	
		1	Indicates a clock request by the BCLKREQ		
3	COM_ MCLK_REQ	0	No clock request by the MCLKREQ	R	
		1	Indicates a clock request by the MCLKREQ		
2	PERIPH_nREQ		Indicates the status of internal peripheral-clock request signal. This is an active-low signal.	R	

Table 38. Status-Request Register (STATUS_REQ_REG) (Continued)

Bit	Name	Value	Description	Type	Reset Value
1	WAKEUP_nREQ		Indicates the status of the internal WAKEUP_nREQ signal. This is an active-low signal.	R	
0	CHIP_IDLE		Indicates the status of the internal CHIP_IDLE signal	R	

The lock-time register (LOCK_TIME) allows fixing the lock time when the APLL is used. It represents the number of CLK (12-MHz or 13-MHz) periods required to activate the APLL lock.

Table 39. Lock-Time Register (LOCK_TIME)

Bit	Name	Description	Type	Reset Value
15−0	LOCK_TIME	Indicates the number of CLK (12-MHz or 13-MHz) periods to wait to activate the lock when the APLL is used. The reset value corresponds to a lock of 200 µs for a 12-MHz CLK.	R/W	0x960

APLL-control register (APLL_CTRL_REG) allows the switch between the APLL and the DPLL. It controls all the input of the APLL.

Table 40. APLL-Control Register (APLL_CTRL_REG)

Bit	Name	Value	Function	R/W	Reset Value
15:4	Reserved		Reserved. These bits should always be written as 0.	R	0xx
3	SEL		Bit used to select the correct divider so that the APLL can generate a 48-mHz clock from either a 12-mHz or 13-mHz reference source. Bit defaults to the 12-mHz reference setting.	R/W	0x1
		0	Divide by 13 provides a 1-mHz clock to the APLL for 48-mHz generation (if the reference clock is 13-mHz)		
		1	Divide by 12 provides a 1-mHz clock to the APLL for 48-mHz generation (if the reference clock is 12-mHz)		

Table 40. APLL-Control Register (APLL_CTRL_REG) (Continued)

Bit	Name	Value	Function	R/W	Reset Value
2-1	RESERVED		Reserved. These bits should always be written as 0.	R/W	0x0
0	APLL_NDPLL_SWITCH		It allows switching between the APLL and DPLL. When 0, use the DPLL; when 1, use the APLL. By default, use the DPLL.	R/W	0x0

Table 41. Power-Control Register (POWER_CTRL_REG)

Bit	Name	Value	Description	Type	Reset Value
15−4	RESERVED		Reserved	R	Unknown
3	SW_NSHUTDOWN		Software generation of the RST_HOST_OUT. This bit controls the state of the RST_HOST_OUT pin when the SW_RST bit is 1.	R/W	0x1
		0	The RST_HOST_OUT is active low		
		1	The RST_HOST_OUT is inactive high		
2	SW_RST		Released hardware generation of the RST_HOST_OUT	R/W	0x0
		0	The state of the RST_HOST_OUT pin depends on the BFAIL/EXT_FIQ and the 32k counter.		
		1	The state of the RST_HOST_OUT pin is equal to the level of the SW_NSHUTDOWN bit.		
1	LOW_PWR_REQ		The low-power software request. When this bit and the LOW_PWR_EN bit are high, the LOW_PWR pin is driven active high.	R/W	0x0
0	LOW_PWR_EN		The low-power enable bit. Disabled by default. This bit enables the usage of the LOW_PWR output pin.	R/W	0x0

Table 41. Power-Control Register (POWER_CTRL_REG)

Bit	Name	Value	Description	Type	Reset Value
		0	During deep-sleep mode, or if the LOW_PWR_REQ is high, the LOW_PWR pin is driven active high.		
		1	The LOW_PWR pin is always driven low.		

Table 42 lists the DSP-idle registers. Table 43 and Table 44 describe the register bits.

Table 42. DSP-Idle Registers

Register Name	Descriptions	R/W	Size	Offset	Reset Value
ICR	DSP-idle-configuration register	R/W	16 bits	x01	0x0000
ISR	DSP-idle-status register	R	16 bits	x02	0x0000

The DSP-idle-configuration register (ICR) indicates the DSP subdomains that are placed in idle mode.

Table 43. DSP-Idle-Configuration Register (ICR)

Bit	Name	Value	Description	Type	Reset Value
15−6	RESERVED				0
5	EMIF_IDLE_DOMAIN	0	No request to idle the DSP EMIF	R/W	0
		1	Request to place the DSP EMIF in the idle mode		
4	DPLL_IDLE_DOMAIN	0	No request to idle the DSP DPLL	R/W	0
		1	Request to place the DSP DPLL in the idle mode		
3	PER_IDLE_DOMAIN	0	No request to idle the DSP peripherals	R/W	0
		1	Request to place the DSP peripherals in the idle mode		
2	CACHE_IDLE_DOMAIN	0	No request to idle the DSP I-cache	R/W	0
		1	Request to place the DSP I-cache in the idle mode		

Table 43. DSP-Idle-Configuration Register (ICR)

Bit	Name	Value	Description	Type	Reset Value
1	DMA_IDLE_DOMAIN	0	No request to idle the DSP DMA controller	R/W	0
		1	Request to place the DSP DMA controller in the idle mode		
0	CPU_IDLE_DOMAIN	0	No request to idle the DSP core and memory	R/W	0
		1	Request to place the DSP core, SARAM, and DARAM in the idle mode		

The DSP-idle-status register (ISR) indicates the DSP subdomains that have been placed in the idle mode.

Table 44. DSP-Idle-Status Register (ISR)

Bit	Name	Value	Description	Type	Reset Value
15−6	RESERVED				
5	EMIF_IDLE_STATUS	0	DSP EMIF is not in idle mode	R	0
		1	DSP EMIF is in idle mode		
4	DPLL_IDLE_STATUS	0	DSP DPLL is not in idle mode	R	0
		1	DSP DPLL is in idle mode		
3	PER_IDLE_STATUS	0	DSP peripherals are not in idle mode	R	0
		1	DSP peripherals are in idle mode		
2	CACHE_IDLE_STATUS	0	DSP I-cache is not in idle mode	R	0
		1	DSP I-cache is in idle mode		
1	DMA_IDLE_STATUS	0	DSP DMA controller is not in idle mode	R	0
		1	DSP DMA controller is in idle mode		

Table 44. DSP-Idle-Status Register (ISR) (Continued)

Bit	Name	Value	Description	Type	Reset Value
0	CPU_IDLE_STATUS	0	DSP core and memory are not in idle mode	R	0
		1	DSP core, SARAM and DARAM are in idle mode		

5 Switching Clock Modes

This section describes the programming guidelines for switching the clock modes in the OMAP5910 device.

5.1 Switching Procedure

Perform the following procedure to switch the OMAP5910 clock modes:

1) Make sure the DSP clock is enabled.
2) Make sure there are no active transfers on interfaces (EMIFS, EMIFF, TIPB, IMIF, MPUI, etc.).
3) Make sure there are no active DSP transactions being performed.
4) Disable the MPU D-cache/MMU.
5) Make sure the MPU-clock-control register (ARM_CKCTL), clock dividers, and the DPLL_REG have correct contents for the clock mode the system is being switched to.
   a) Switch from SYNC mode to SYNCSCALE mode:
      i) Make sure that the frequency of the traffic controller is always less than the maximum frequency of the traffic controller.
      ii) Change the clock mode.
      iii) Program the clock dividers.
      iv) Program the DPLL to the frequency desired.
   b) Switch from SYNCSCALE to SYNC mode:
      i) Program the DPLL to the desired frequency in synchronous mode.
      ii) Program all clock dividers to be equal.
      iii) Change the clock mode to SYNC mode.

6) After the MPU write to the MPU-system-status register (ARM_SYSST) (0x18) to switch modes, there must be no requests from the MPU to the traffic controller for the next 100 MPU cycles (see Section 5.2, *Main Code*, and Section 5.3, *Delay Procedure*).

7) Make sure all read and write accesses to the clock-reset registers are 16-bit accesses.

5.2 Main Code

The following is the main code for switching modes:

```
main()
{
   …
…..
// Enable Icache
   INT_SetSupervisor();
   ARM_WRITE_REG1(I_bit);
   INT_SetUser();
// Enable DSP Clock
   MCU_CKCTL = 0x2000;
   switch_mode(CLOCK_MODE_SYNC_SCALE);
// Passing in 0x1000
….
….
}
```

5.3 Delay Procedure

Use the following software routine to create a delay of 100 clock cycles after a switch-mode write has occurred.

Ensure the I-cache is enabled during the switching modes.

```
      state16               ; thumb mode
      .ref  edata           ; defined by armas
      .global $switch_mode
$switch_mode:
       push    {lr}
       push     {r1-r7}
       adr      r4, into_32_bis
       bx       r4
   nop
   nop
   nop
       .state32                ; arm mode
into_32_bis:
       ;
      LDR    R1,ARM_SYSST
      MOV    R3,#0
      MOV    R2,#0
; This is the loop that will wait for at least 100 cycles
; before issuing next request from MPU. On the first run
of the loop only Icache
; gets loaded with the loop and the next 2 instructions
but write to SYSST does not occur
; In the 2nd run of the loop only write to SYSST happens
and after that MPU runs the loop from
;Icache so no request goes out
LOOP     CMP    R2,#1
         STREQ  R0,[R1]
         ADD    R2,R2,#1
         CMP    R2,#16
         BNE    LOOP
the_end:
         adr    r2, into_16_bis + 1
         bx     r2
   .state16
into_16_bis:
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       pop   {r1-r7}
       pop   {pc}
;**********************************************************
;* CONSTANT TABLE                                         *
;**********************************************************
ARM_SYSST      .long 0xFFFECE18
```

Appendix A

Revision History

This document was revised to SPRU678A from SPRU678, which was released in October 2003. The scope of the revisions was limited to technical changes as described in A.1. This appendix lists only revisions made in the most recent version.

A.1 Changes Made in This Revision

The following changes were made in this revision:

Page	Additions/Modifications/Deletions
83	Added information on how to switch from DPLL Bypass to DPLL Lock Mode and vice−versa as sections 4.1.1 and 4.1.2

B Index