OMAP5912 Multimedia Processor OMAP3.2 Subsystem Reference Guide

Literature Number: SPRU749B
April 2006

OMAP™
TEXAS INSTRUMENTS TECHNOLOGY

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Preface

Read This First

About This Manual

This document introduces and briefly defines the main features of the OMAP3.2 subsystem of the OMAP5912 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

Documentation that describes the OMAP5912 device, related peripherals, and other technical collateral, is available in the OMAP5912 Product Folder on TI's website: www.ti.com/omap5912.

Trademarks

OMAP and the OMAP symbol are trademarks of Texas Instruments.

Contents

Figures

Tables

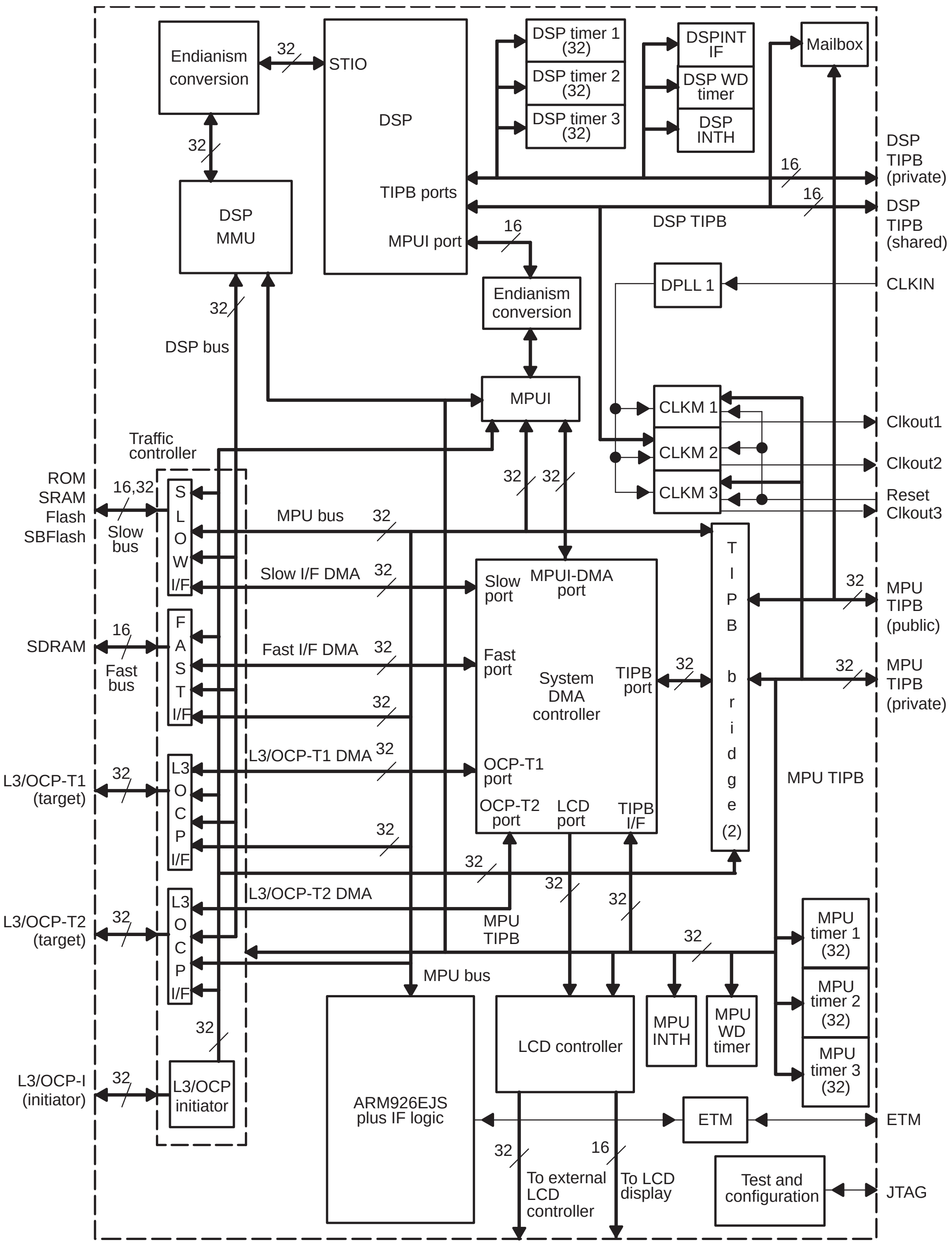
OMAP3.2 Subsystem

This document introduces and briefly defines the main features of the OMAP3.2 subsystem of the OMAP5912 multimedia processor.

1 Introduction

Figure 1 shows the OMAP5912 OMAP3.2 gigacell.

Figure 1. OMAP3.2 Gigacell



2 OMAP3.2 Features

The OMAP3.2 includes the following features:

- ARM926EJS megacell, including:
  - ARM926EJS, running at a maximum frequency of 192 MHz
  - MMU with translation lookaside buffer (TLBx)
  - L1 16K-byte, four-way, set-associative instruction cache
  - L1 8K-byte, four-way, set-associative data cache with write buffer
- MPU core interrupt handler level 1
- Embedded trace macrocell module, ETM version 2.a, in a 13-bit mode configuration or in a 17-bit demultiplexed mode configuration
- DSP core megacell rev 2.0a+, including:
  - Embedded ICE emulator interface through JTAG port
  - TMS320C55x (C55x) DSP core rev 2.1, running at a maximum of 192 MHz
  - L1 cache (24K bytes)
    - 16K-byte, two-way, set-associative instruction cache *(on the OMAP5912 prototype, one wait state is introduced in case of discontinuity)*
    - 2 X 4K-byte RAM set for instruction
  - DARAM 64K-byte, zero wait state, 32-bit organization
  - SARAM 96K-byte, zero wait state, 32-bit organization
  - PDROM (32K bytes)
  - DSP/System DMA controller: six physical channels, five ports
  - DSP core trace module
  - Hardware accelerators motion estimation (ME), discrete/inverse discrete cosine transform (DCT/IDCT), and pixel interpolation (PI)
  - DSP core interrupt handler level 1 in the C55x DSP core
- DSP core MMU
- DSP core level 2 interrupt handler, which enables connection to 16 additional interrupt lines outside OMAP. The priority of each interrupt line is controlled by software.

- DSP core interrupt interface, which enables connection to the interrupt lines coming out of the level 2 interrupt handler and the interrupt lines requiring more priority. The outcome interrupt of this module is then connected to the DSP core megacell to be processed by the DSP core. This module mainly ensures that all interrupts going to the DSP core megacell are level-sensitive.
- DSP core peripherals:
  - 3 x 16-bit DSP core private timers
  - 1 x 16-bit DSP core private watchdog
- Mailboxes:
  - Four mailboxes are implemented:
    - Two read/write accessible by MPU core, read-only by the DSP core
    - Two read/write accessible by the DSP core, read-only by the MPU core

    Each mailbox is implemented with 2 x 16-bit registers. When a write is done into a register by one processor, it generates an interrupt, released by the read access of the other processor.
- MPU core peripherals
  - 3 x 32-bit private timers; their clock is either the OMAP3.2 reference input clock or the divided MPU core clock.
  - 1 x 16-bit private watchdog; can be configured as a 16-bit general-purpose timer by software. Its clock is the OMAP3.2 reference input clock divided by 14.
- External LCD controller support in addition to the OMAP LCD controller
  - LCD controller with its own tearing-effect logic
- Memory traffic controller
  - External Memory Interface Fast (EMIFF) is a memory interface that enables 16-bit data SDRAM memory access at 96-MHz maximum frequency. It supports connection to a128M-byte maximum of SDRAM. The address width is 16 bits, and two bank selection bits are also provided. The OMAP5912 chip requires interfacing with a maximum of four banks of 64M x 16-bit SDRAM memory with DDR capability.

  - External Memory Interface Slow (EMIFS) connects external device memories (such as common flash and SRAM memories) at 80-MHz maximum frequency. This interface is also used internally to connect the boot ROM. This interface enables 16-bit data accesses and provides four chip-selects. Each chip-select is able to support up to 64M bytes of address space through a 25-bit address bus.

Note:

At the OMAP5912 level, two chip-selects can be split in half by configuration to provide four chip-selects. This enables OMAP5912 to provide up to six chip-selects, supporting up to 32M bytes of address space on four chip-selects and up to 64M bytes on two chip-selects.

L3 OCP-T1 and L3 OCP-T2 ports are provided to enable memory access from the OMAP3.2 gigacell on a standard basis protocol. Only the L3 OCP-T1 is used in OMAP5912 to access the single SRAM memory.

- ❑ Emulator interface through JTAG port

- ❑ System DMA running at 96 MHz. It consists of:

  - Seventeen logical channels
  - Seven physical ports + one for configuration
  - Four physical channels

  The ports are connected to the L3 OCP targets, the external memory, the TIPB bridge, the MPUI, and one dedicated port connected to an LCD controller. The system DMA controller can be controlled via the MPU core private TIPB or by an external host via the OCP-I port.

  The system DMA controller is designed for low-power operation. It is partitioned into several clock domains where each clock domain is enabled only when it is used. All clocks are disabled when no DMA transfers are active, synchronous to the MPU core TIPB. This feature is completely under hardware control; no specific programming is needed.

  Five different logical channels types are supported, each one representing a specific feature set:

  - LCh−2D for memory to memory transfers, 1D and 2D
  - LCh−P for peripheral transfers
  - LCh−PD for peripheral transfers on a dedicated channel
  - LCh−G for graphical transfers/operations
  - LCh−D for display transfers

The available features are:

- Support for up to four address modes:
  - Constant
  - Post-increment
  - Single indexing
  - Double indexing
- Different indexing for source-respective destination
- Logical channel chaining
- Software enabling
- Hardware enabling
- Logical channel interleaving
- Logical channel preemption
- Two choices of logical channel arbitration of physical resources, round robin or fixed
- Two levels of logical channel priority
- Constant fill
- Transparent copy
- Rotation 0, 90, 180, and 270
- Seven ports enabling:
  - Memory-to-memory transfers
  - Peripheral-to-memory transfers
  - Memory-to-peripheral transfers
  - Peripheral-to-peripheral transfers
- Binary backward-compatible by default configuration
- Up to four logical channels active in parallel

The logical channel dedicated to the display, LCh−D, has several additional features:

- Channel can be shared by two LCD controllers
- Supports both single- and dual-block modes
- Supports separate indexing and numbering for dual-block mode for both elements and frames

❑ Two DPLLs:

OMAP3.2 provides one DPPL per main clock domain:

- MPU core/traffic controller clock domain
- DSP core clock domain

The OMAP3.2 gigacell enables the software to define either:

- ■ Two coupled domains in scalable mode: only one DPLL is active. The other clocks are a multiple of it.
- ■ Mixed mode: only one domain is working in asynchronous mode. The other domains are in scalable mode.

❑ Endianism conversion for DSP core

- ■ The DSP core uses big-endian format, whereas the MPU core uses little-endian format. Also, as a rule, the OMAP5912 chip works in little endian. Thus, the endianism conversion is useful for all memory or peripheral accesses from on-chip peripherals or all shared memories to the DSP core megacell.

OMAP 3.2 is considered a subchip of OMAP5912. To connect the OMAP peripherals, six buses are provided:

- ❑ MPU core shared TIPB
- ❑ MPU core private TIPB
- ❑ DSP core shared TIPB
- ❑ DSP core private TIPB
- ❑ OCP-T2
- ❑ OCP-I

3 Traffic Controller

The OMAP 3.2 traffic controller (TC) is the central interconnect that manages all accesses between the following:

- ❑ OMAP internal initiators and target resources.
- ❑ OMAP external initiators and target resources.

The TC can have its own clock domain or be synchronous to the MPU core clock domain. See the *OMAP5912 Multimedia Processor Clocks Reference Guide* (SPRU751)*,* for more details on clock domains. Typically, the TC clock runs at half of the MPU core and DSP core clocks.

System initiators are:

- ❑ MPU core*.* The MPU core is connected to the TC using the MPU core bus. The MPU core can access memory devices or other type of targets connected to OCP-T1, OCP-T2, EMIFF, and EMIFS. Selection of the destination target is based on address decoding within the TC.
- ❑ DSP core*.* The DSP core is connected to the TC via the DSP core bus. The DSP core can access memory devices or other types of targets connected to OCP-T1, OCP-T2, EMIFF, and EMIFS. Selection of the destination target is based on address decoding within the TC.

- ❑ System DMA. The system DMA is connected to the TC with four dedicated ports: one for OCP-T1, one for OCP-T2, one for EMIFF, and one for EMIFS. It can access memory devices or other types of targets connected to any of these interfaces. Selection of the destination port is based on system DMA programming.

- ❑ External initiator. An external initiator can be connected to the TC via the OMAP OCP-I port. The external master can access memory devices or other types of targets connected to OCP-T1, OCP-T2, EMIFF, and EMIFS.

System targets are:

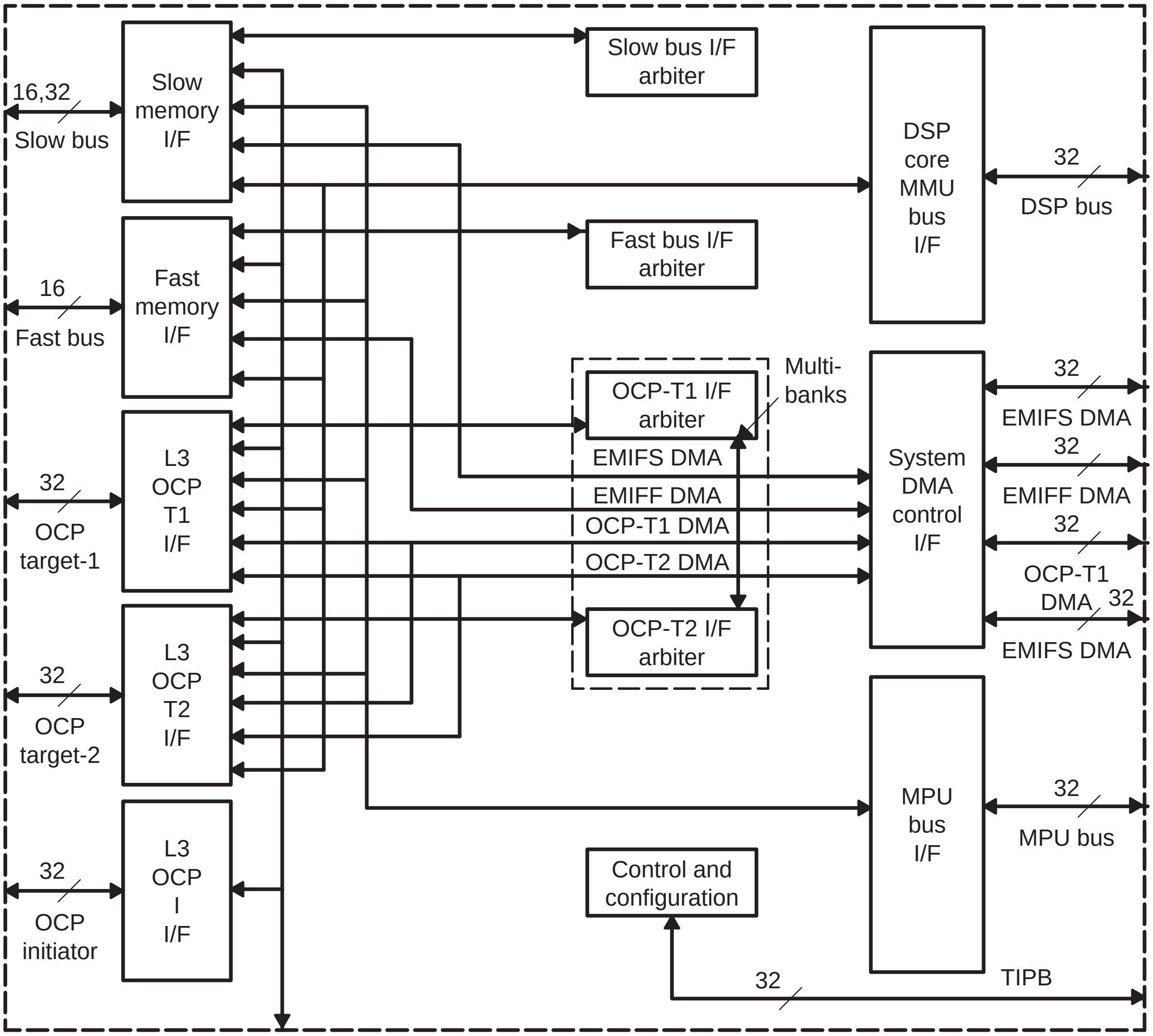
- ❑ EMIFS memory interface. This memory controller supports most common memory interface protocols through a flexible programming and timing control that also allows support of any type of internal or external IC target module.
  - ■ External memory devices like asynchronous ROM, RAM, flash, or synchronous burst flash.
  - ■ Internal TI ACE ROM, RAM memories.
  - ■ Internal and external asynchronous target modules.

- ❑ EMIFF memory interface. This memory controller supports most common SDRAM interface protocols through a flexible programming and timing control.
  - ■ External synchronous standard single-data rate (SDR SDRAM) and low-power SDR SDRAM.
  - ■ External synchronous standard double-data rate (DDR SDRAM) and mobile DDR SDRAM.

- ❑ OCP-T port interface. OMAP includes two ports on which OCP-compliant slaves can be connected. These two target ports are called OCP-T1 and OCP-T2. Attached target modules can be either internal or external memory subsystems or any type of target peripheral.

Figure 2 shows the traffic controller functional block diagram:

Figure 2. Traffic Controller Functional Block Diagram



3.1 OCP-T1/OCP-T2

The OMAP 3.2 hardware provides two identical general-purpose ports for the connection of custom peripherals or memory subsystems. These ports can be accessed by all the initiators mentioned in the previous sections.

Each port arbitrates between all initiator requests and allows atomic transactions for the MPU core, the DSP core, and the external initiator to implement semaphore-based synchronization schemes between software tasks (running either on the same CPU or on two different CPUs).

Once granted, the selected request is converted from an OMAP internal protocol into an OCP-compliant transaction. As seen at the OMAP boundary, the OCP-TX is an OCP master.

OCP-T1 and OCP-T2 have their own addressing spaces within the OMAP memory map (256M bytes accessible by OCP-T1 and 1.2 gigabytes accessible by OCP-T2) plus an additional common address range called the multibank address space (256M bytes). The multibank address space is used in some devices for connecting a dual-port OCP slave, such as a memory controller, that can provide concurrent data flow to/from memory banks.

A request that targets the multibank address space is analyzed and can be routed either to OCP-T1 or OCP-T2 by the traffic controller.This mechanism is transparent to the programmer.

OCP-T1 and OCP-T2 both implement the arbitration schemes described in section 3.6, *Priority Algorithms*.

Bus error transactions that generate abort signals on the external OCP interconnect can be tracked within the OCP-T1/T2 controllers.

Two types of errors can be detected:

- Bus error returned by the OCP interface.
- Time-out error. A request has been sent to the interface but has not been completed in the time-out delay. The timer is programmable from 0 to 255 TC clock cycles.

The abort interrupt handler can read the error reporting registers, which indicate the address that caused the error, the initiator at the origin of the request, and the type of error (see abort address register and abort type register, in section 3.5.4).

3.2 EMIFS Programming

The synchronous/asynchronous external memory interface slow (EMIFS) supports most common memory interface protocols through a flexible programming and timing signals control.

- The EMIFS can control up to six devices without adding any external logic through the four independent chip-selects ($\overline{\text{FLASH.CS0−3}}$) (plus two optional chip selects within CS1 and CS2) and through dedicated memory interface control signals.
- The EMIFS supports common external memory control signals, such as OE, WE, ADV, BE[0−1], BAA, Ready, Device CLK, RST, and WP.

- Each chip-select controls an address range of 64M bytes with dedicated configuration registers to fulfill the protocol and the timing constraints compliant with the external device associated with it. Each chip-select configuration register supports dynamic configuration.

- CS0 and CS3 address mapping can be swapped with the BM bit in the EMIFS configuration (EMIFS_CONFIG) register.

- The EMIFS can support 16-bit interface width only. Based on the CS configuration, the interface adjusts the access size (splitting word32) according to external device attached to the CS. **User must refer to the IC device documentation for the external device width supported by the IC (16-bit width only or 16-bit and 32-bit width).** 8-bit device width is not supported without adding external logic.

- The EMIFS can control multiplexed address and data memory devices without adding external logic based on CS configuration. The multiplexing scheme is supported for synchronous and asynchronous access mode. Both multiplexed and non-multiplexed devices can be supported with the same IC on different chip-selects (embedded IC non-multiplexed memories and external multiplexed devices).

- The EMIFS behavior conforms to the little endian protocol.

- The EMIFS supports 8-,16-, or 32-bit asynchronous and synchronous read, 4- x 32-bit synchronous burst read and 8-, 16-, or 32-bit asynchronous write.

- The EMIFS boot configuration is controlled by dedicated pins that are sampled at IC reset time. This provides flexible boot CS and configuration selection.

- The EMIFS is a multimaster memory interface. It supports flexible and programmable arbitration protocol (LRU priority ordering or dynamic time based priority ordering).

- The EMIFS includes a programmable time-out timer to prevent system hanging with nonresponding devices. Automatic access completion with interrupt and status logging are issued on time out events.

- The EMIFS supports dynamic local idle mode control. The EMIFS also supports IC deep power-down mode request synchronization.

3.2.1 General Description

EMIFS Synchronous and Asynchronous Modes

- ❑ The operation mode of the EMIFS for a given chip-select region is selected by the RDMODE bit field of the CS configuration registers. Operations supported are:
  - ■ Mode 0. Asynchronous read. Used for any asynchronous memory, including flash devices.
  - ■ Mode 1−2−3. Asynchronous page mode read with control of 4 (mode1), 8 (mode 2), and 16 (mode 3) words (device width) per page. These modes are mainly used for page mode flash devices.
  - ■ Mode 4−5. Synchronous burst read, with burst advance control for mode 4. These modes are mainly used for synchronous burst flash devices.
  - ■ Mode 7. Synchronous pipelined burst read. This mode is mainly used for TI embedded IC ROM and RAM memories.

  For all these modes, write accesses are performed according to asynchronous write protocol.
- ❑ Single and burst access address alignment
  - ■ OMAP master only issues Word16 and Word32 aligned access (word address must be aligned on word size address boundary).
  - ■ OMAP master only issues linear, incrementing, and fixed size 4xWord32 access bursts. Burst access is aligned on burst size address boundary. Starting burst LSB address A[0−3] is always equal to [0000]. External devices like synchronous flash memory may require a burst protocol programming to conform to the EMIFS burst protocol.

EMIFS Memory Timing Control

- ❑ In both asynchronous and synchronous modes, all EMIFS to memory control signals are controlled with an EMIFS internal reference clock (REF_CLK) which is the TC_CK divided down. Depending on the CS configuration, this internal clock can be available outside through the FLASH.CLK (ball N3) output pin.
  - ■ The REF_CLK is divided from TC_CK (traffic controller clock, see section 4.2.5) by a programmable value contained in FCLKDIV bit field of the CS configuration register, Table 19. This accommodates the timing constraints of slow devices, even with high system clock rate.

- In asynchronous mode 0−1−2−3, the FLASH.CLK (ball N3) is low.
- In synchronous mode 4−5−7, REF_CLK is available through FLASH.CLK. This is also the case during asynchronous write access. FLASH.CLK clock is connected to the external synchronous device input clock (synchronous flash or ASIC). The frequency must be set to comply with the attached device's timing constraints in combination with the others EMIFS timing controls.
- In synchronous mode 4−5, a retiming mode enables read data to be latched by a delayed REF_CLK. This retiming mode must not be used with asynchronous modes.
  - The REF_CLK delay is obtained through the IC I/O feedback of FLASH.CLK to offer optimum data and clock alignment. Pipelined access offers relaxed timing constraints.
  - The retiming mode is enabled through RT bit field in the CS configuration register (see Table 19).
- Memory control signal (CS, OE, WE, ADV, ADDRESS) setup and hold timing can be controlled by programmable internal delay generation.
  - OE valid/invalid timing from/to CS and address valid/invalid is programmable through the OESETUP, OEHOLD bit fields in the advanced CS configuration register.
  - WE valid timing from CS, address, and data valid is programmable through WRWST bit field in the CS configuration register. WE hold time is fixed to one REF_CLK.
  - ADV valid pulse time is programmable through ADVHOLD bit field in the advance CS configuration register.
    - In synchronous read mode, the ADV setup to REF_CLK rising edge is controlled by the ADVHOLD bit field. The ADV hold time from REF_CLK rising edge is fixed to one TC_CK.
  - Address setup and hold time with ADV control:
    - In synchronous mode, the address setup time from REF_CLK rising edge (ADV valid) is controlled by ADVHOLD bit field.
    - In multiplexed synchronous mode, the address hold time from REF_CLK rising edge is fixed to one REF_CLK from ADV invalid time.
    - In asynchronous mode, the address setup time to ADV invalid is controlled by ADVHOLD bit field.
    - In asynchronous mode, address hold time from ADV invalid is fixed to one REF_CLK.

- ❑ Read and write access time is controlled by programmable internal wait state generation (non-full-handshaking mode). An external Ready input pin (FLASH.RDY on ball V2) can also be used in combination with internal wait state (full-handshaking mode).
  - ■ In non-full-handshaking mode, the RDWST and PGWST (mode 1−2−3 only) bit fields in the CS configuration register are used to control internal read wait state generation.
  - ■ In non-full-handshaking, the WELEN bit field in CS configuration register is used to control internal write wait state generation.
  - ■ In full-handshaking mode, FLASH.RDY is monitored by the EMIFS to control read and write access time. The access is completed when both the internal wait state has expired and FLASH.RDY is asserted by the external device. The FLASH.RDY assertion/deassertion timing constraint depends on synchronous or asynchronous access mode.
  - ■ Modes 0−4−5 are by default in full-handshaking mode. Full-handshaking support on a particular CS can be disabled for these modes through the dynamic wait state register.

    Modes 1−2−3−7 always follow the non-full-handshaking protocol and FLASH.RDY is never monitored in these modes, even if the full-handshaking bit field in the dynamic wait state register is cleared.
- ❑ To prevent data bus contention when slow devices are attached to the IC, the BTWST bit field in the CS configuration register is used to control TC_CK cycle idle time between specific access sequences.
- ❑ The BTMODE field in the advance CS configuration register extends the previous mode. The BTWST bit field controls the CS negation time between successive accesses to the same CS.
- ❑ To prevent data bus floating when no access is requested at the interface (idle sequence), the EMIFS keep the data bus driven with the previous written data or read data (bus keeping feature). In case of read to idle sequence, the delay time from read to write-back is at least one TC_CK cycle or controlled by BTWST (TC_CK) cycles.
- ❑ If dynamic Wait state mode is used, then one REF_clk cycle must be added to all of the formulas describing CS and ADV width in this section.

3.2.2 EMIFS CS0 and CS3 Decoding Control

CS0 and CS3 address decoding (address in the TC memory mapping) can be swapped through the BM bit field in the EMIFS global control register. When the BM bit field is set, CS3 is activated in the 0000:0000−03FF:FFFF range and CS0 is activated in the 0C00:0000−0FFF:FFFF range. The BM bit is sampled at reset, depending on two factors. If MPU_BOOT (ball J20) is 1 and the device type is emulation, then BM resets to 1. Otherwise, BM resets to 0. Thus, the boot is executed from CS0 or from CS3 attached memories. During normal execution, BM can be changed dynamically, but obvious software precautions are required to prevent system crash.

3.2.3 EMIFS Miscellaneous Memory Signal Control

Common flash memory supports the write protection, WP, input pin ($\overline{\text{FLASH.WP}}$ on ball V4) to prevent erroneous write access sequence from corrupting their content. EMIFS interface includes a WP output pin with full software control.

Common flash memory supports the RESET input pin to allow device state machine to be properly reset on power up, and also to minimize power consumption in idle mode. The EMIFS interface includes a $\overline{\text{FLASH.RP}}$ (on ball W1) output pin with software and hardware control.

3.2.4 EMIFS Configuration

EMIFS control and configuration can be done dynamically. EMIFS ensures that a new CS configuration takes effect only when no access is requested (CS idle). To prevent inconsistency and critical behavior, the EMIFS configuration must be done by MPU core software while other masters are inactive.

3.2.5 EMIFS Abort Control

In the case of access restriction violations, or access time-out errors, the EMIFS can handle an interrupt line and abort status register to allow abort events system monitoring.

3.2.6 EMIFS External Device Connections

Please refer to *OMAP5912 Multimedia Processor Memory Interfaces Reference Guide* (SPRU756) for a detailed description of the EMIFS device connections.

3.2.7 EMIFS Address Mapping and Data Control in Multiplexed Mode

- ❏ In order to minimize the number of IC pins for external memory connection, the EMIFS can support multiplexed address and data memory devices without adding external logic.
- ❏ Selection of the multiplexed mode is done through the MAD bit in the CS configuration register. Multiplexed mode is only available in the following modes:
  - ■ Mode 0 asynchronous read and write
  - ■ Mode 4 & mode 5 synchronous burst read

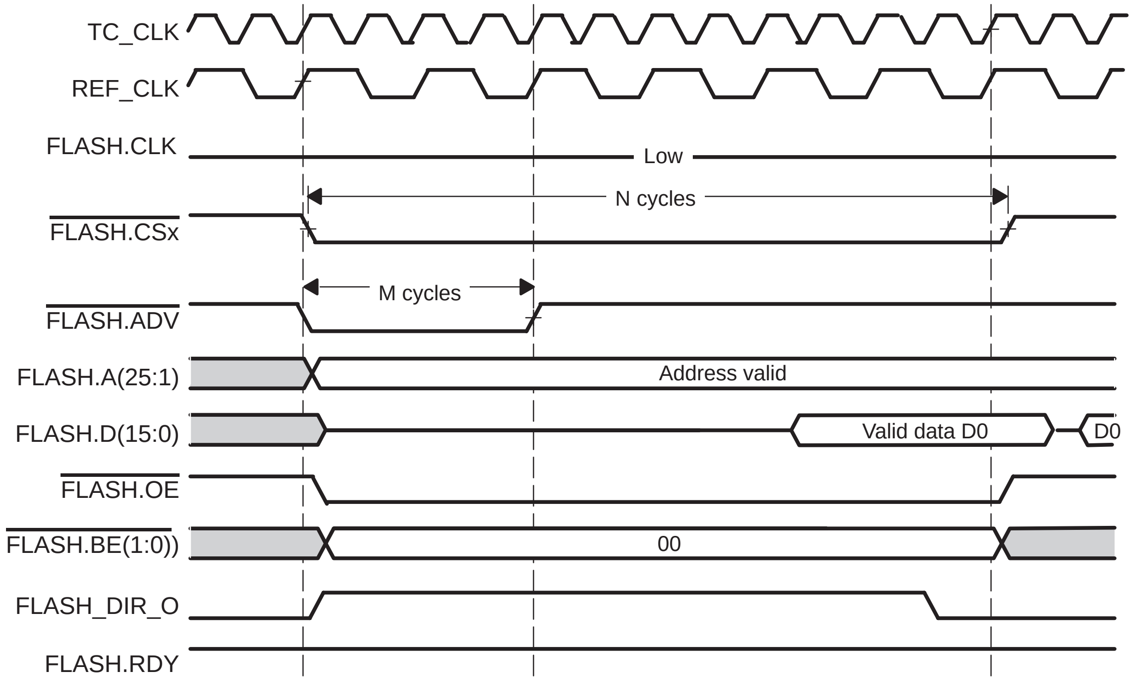
3.2.8 Mode 0—Asynchronous Read Operation

Basic Programming Model

The asynchronous read mode 0 is selected by setting RDMODE = 0 in the corresponding EMIFS chip-select configuration register.

Figure 3 shows a typical timing diagram.

Figure 3. Asynchronous 16-Bit Read Operation on a 16-Bit Width Device. RDWST=4, FCLKDIV=1, OESETUP = 0, OEHOLD = 0, ADVHOLD = 1. Data write-back on the bus after read completion.

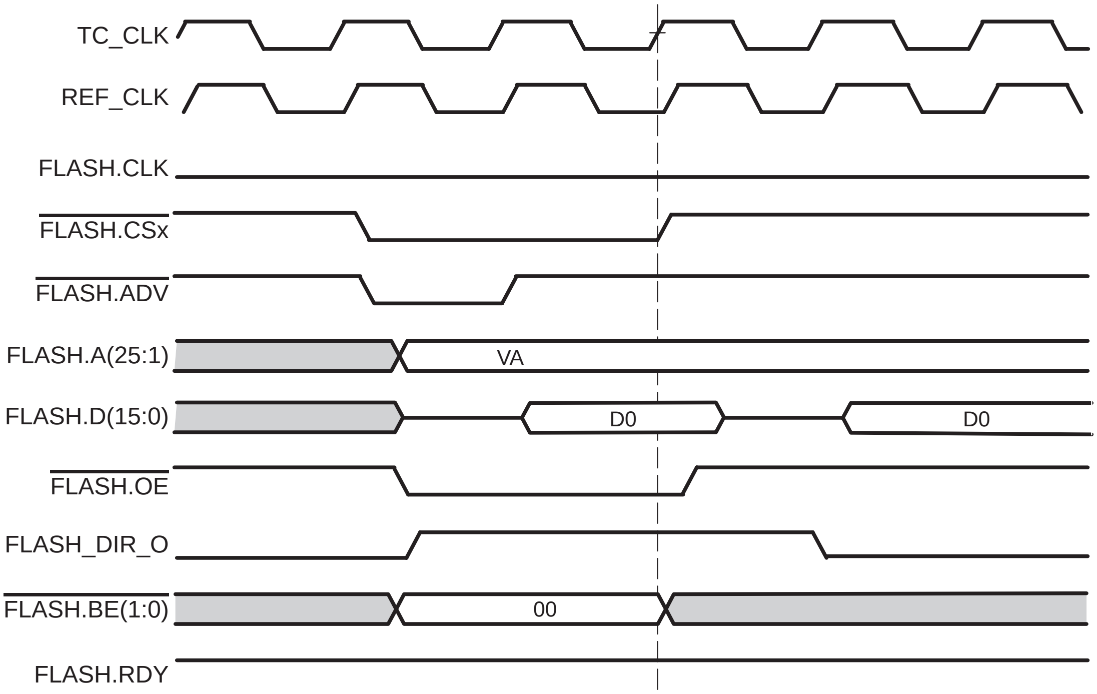


- ❑ The REF_CLK is divided from TC_CK (traffic controller root clock) by a programmable value contained in the FCLKDIV bit field of the CS configuration register.

FCLKDIV	REF_CLK
00	TC_CK:1
01	TC_CK:2
10	TC_CK:4
11	TC_CK:6

- ❑ The CS pulse width depends on RDWST bit field of the CS configuration register. CS pulse width equals:
  - ■ (RDWST + 2) REF_CLK (N cycles in Figure 3).
  - ■ CS minimum pulse width is 2 REF_CLK.
- ❑ The ADV pulse width depends on the ADVHOLD bit field of the CS configuration register. ADV pulse width equals:
  - ■ (ADVHOLD + 1) REF_CLK (M cycles in Figure 3).
  - ■ ADV minimum pulse width is 1 REF_CLK.
- ❑ Address drive time follows CS activation (no setup time guaranty). Address setup time to ADV rising edge is controlled by ADVHOLD. Address hold time from ADV rising edge is controlled by CS pulse width.
- ❑ Read data is latched on the same TC_CK rising edge that deactivates OE signal.
- ❑ After a read completion, if no other access (RD, WR) is pending, the data bus is driven with the previous read value. The bus turn-around time (OE going high to direction going out) is a minimum of 1 TC_CK cycle and can be extended through BTWST.
- ❑ In asynchronous mode, REF_CLK is not provided outside the EMIFS and Flash_clk_o is kept low.

Figure 4. Asynchronous 16-Bit Read Operation on a 16-Bit Width Device. RDWST=0, FCLKDIV=0, OESETUP=0, OEHOLD=0, ADVHOLD=0. Data write-back on the bus after read completion.



Advanced OE Control

❑ OE activation delay time from CS and address valid is programmable through the OESETUP bit field in the advanced CS configuration register. Activation delay timing is equal to:

- (OESETUP) REF_CLK (OESETUP cycles in Figure 5).

❑ OE deactivation advance time to CS and address invalid is programmable through the OEHOLD bit field in the advanced CS configuration register. Deactivation advance timing is equal to:

- (OEHOLD) REF_CLK (OEHOLD cycles in Figure 5).

❑ Because CS minimum pulse width is 2 REF_CLK, the OE delay and advance timing value must be set so that (OESETUP + OEHOLD) is inferior or equal to RDWST. Noncompliant programming ends up with bad access completion.

Figure 5. Asynchronous 16-Bit Read Operation on a 16-Bit Width Device. RDWST=4 FCLKDIV=1 OESETUP=3 OEHOLD=0 ADVHOLD=0. Data write-back on the bus after read completion.

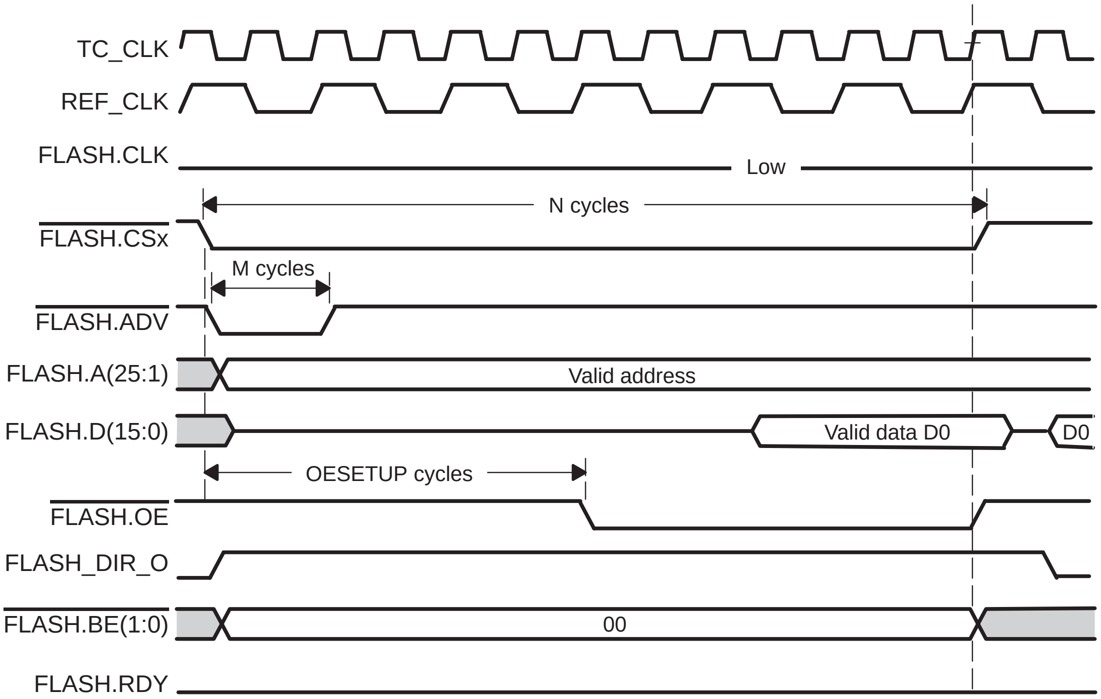
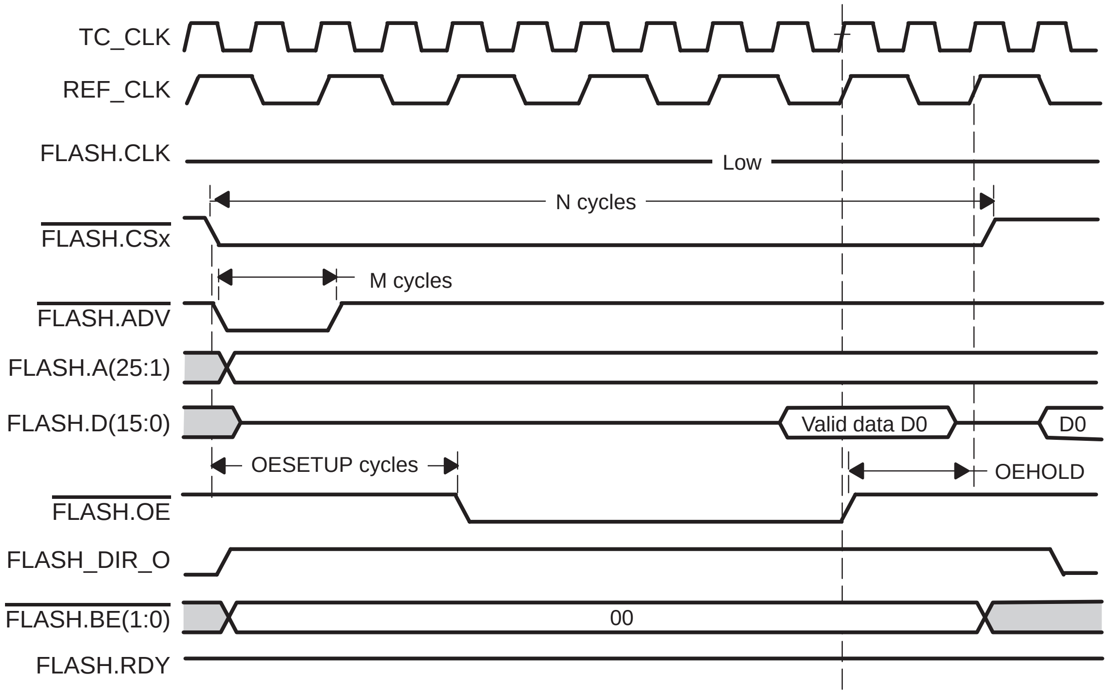


Figure 6. Asynchronous 16-Bit Read Operation on a 16-Bit Width Device. RDWST=4 FCLKDIV=1 OESETUP = 2 OEHOLD = 1 ADVHOLD = 0. Data write-back on the bus after read completion.



Read Access Size Adaptation and CS Pulse Width High Control

- ❑ In read mode 0, the EMIFS splits the Word32 access into two Word16 accesses in case of 16-bit device width. 4xWord32 burst reads are split into eight successive Word16 accesses. The split process follows the little endian protocol (Word32 LSB part at lower Word16 address).
- ❑ During split read accesses and during burst read accesses, the CS signal is deactivated for at least one TC_CK between two successive accesses. CS pulse-width high time can be extended by the BTWST field in the CS configuration register (see also bus turn around and CS negation time control).
  - ■ CS pulse width high = (BTWST +1) TC_CK

Figure 7. Asynchronous 32-Bit Read Operation on a 16-Bit Width Device. RDWST=4 FCLKDIV=0 OESETUP = 0 OEHOLD = 0 ADVHOLD = 0 BTWST=0 BTMODE=0. Data write-back on the bus after read completion.

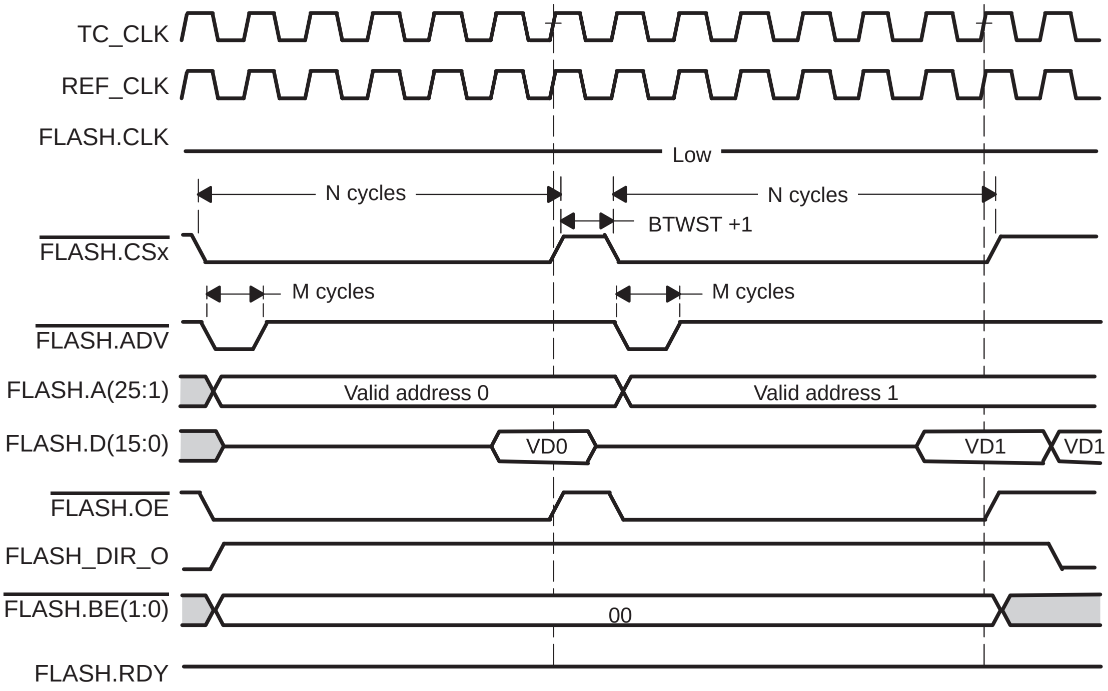
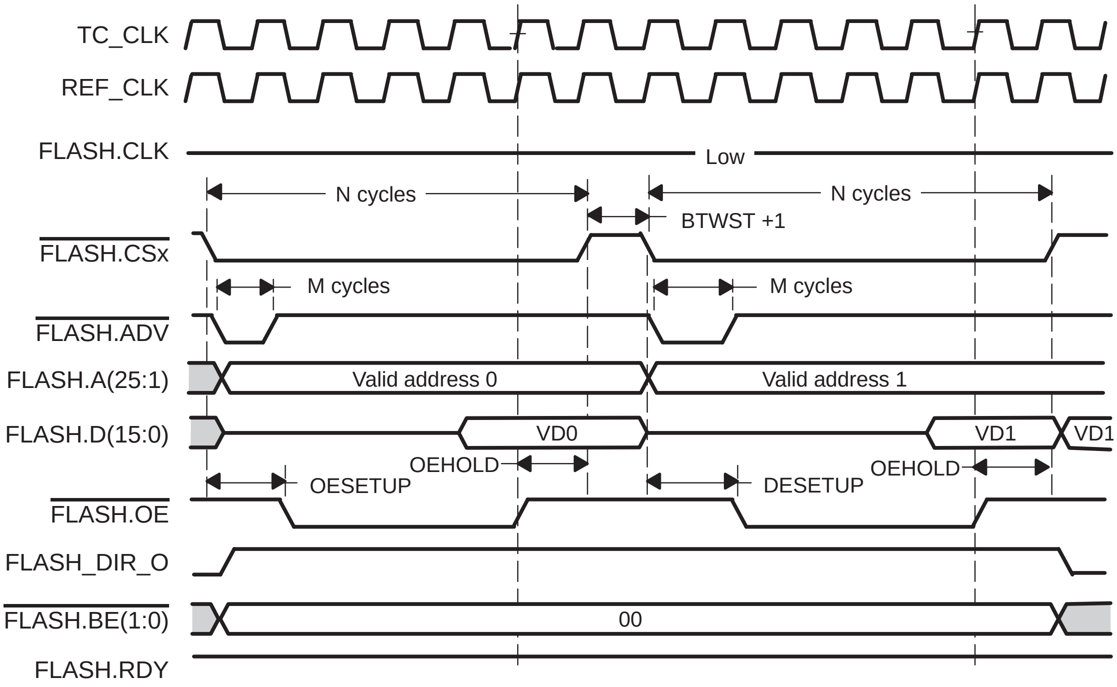


Figure 8. Asynchronous 32-Bit Read Operation on a 16-Bit Width Device. RDWST=4 FCLKDIV=0 OESETUP = 1 OEHOLD = 1 ADVHOLD = 0 BTWST=0 BTMODE=0. Data write-back on the bus after read completion.



Full-Handshaking and Ready Pin Usage in Asynchronous Read Mode

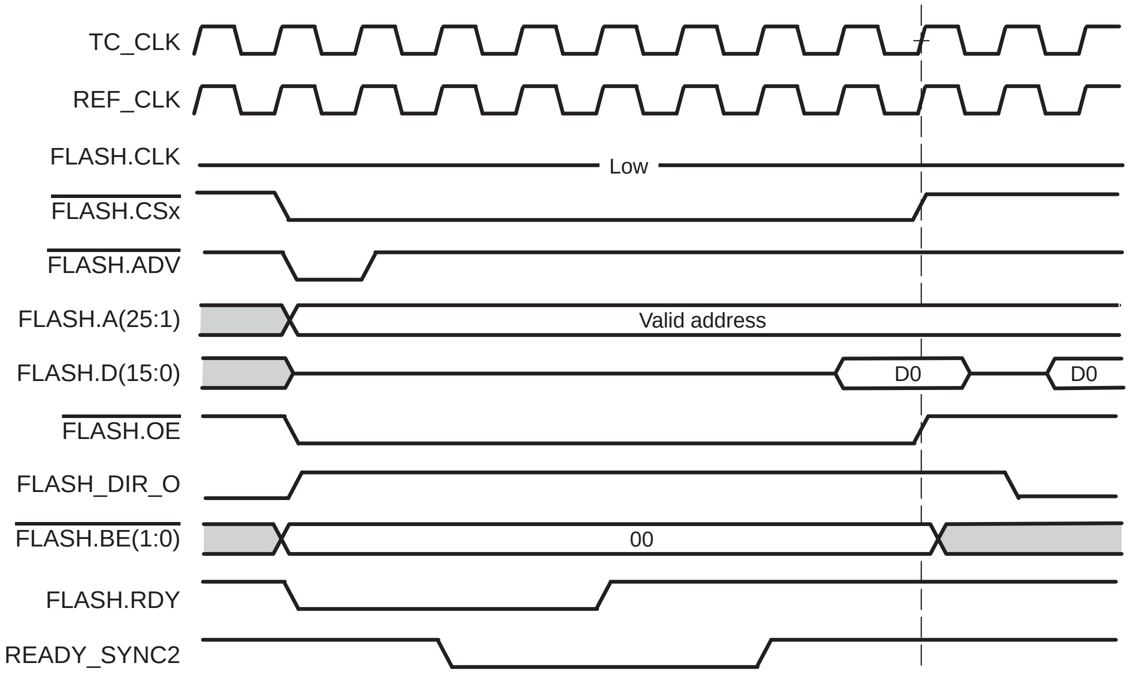
- In full-handshaking mode, the READY input pin (FLASH.RDY on ball V2) is monitored by the EMIFS to control read access time. The access is completed when internal wait state RDWST expires and FLASH.RDY is asserted by the external device.

- ADV pulse width time and OE assertion time are not dependent on the FLASH.RDY pin state.

- When FLASH.RDY is used to extend the access time, the access completion is not controlled by internal delay generation. CS, OE, and address are deactivated when FLASH.RDY is detected high. No OEHOLD time can be control in this case, and the bit field must be equal to zero.

- ❑ Since FLASH.RDY is an asynchronous signal, a nonready device must drive FLASH.RDY low enough time ahead of the minimum access time completion. Depending on FLASH.RDY assertion low delay from CS active, and depending on REF_CLK frequency, a minimum RDWST value may be needed for the FLASH.RDY state to be correctly monitored by the EMIFS.
  - ■ As an example, a minimum of RDWST = 2 is needed for a nonready device that drives FLASH.RDY low with 0 time delay from CS low and for a CS configuration FCLKDIV = 0.
  - ■ See the *OMAP5912 Data Manual* (SPRS231) for timing information regarding the FLASH.RDY assertion timing constraint.

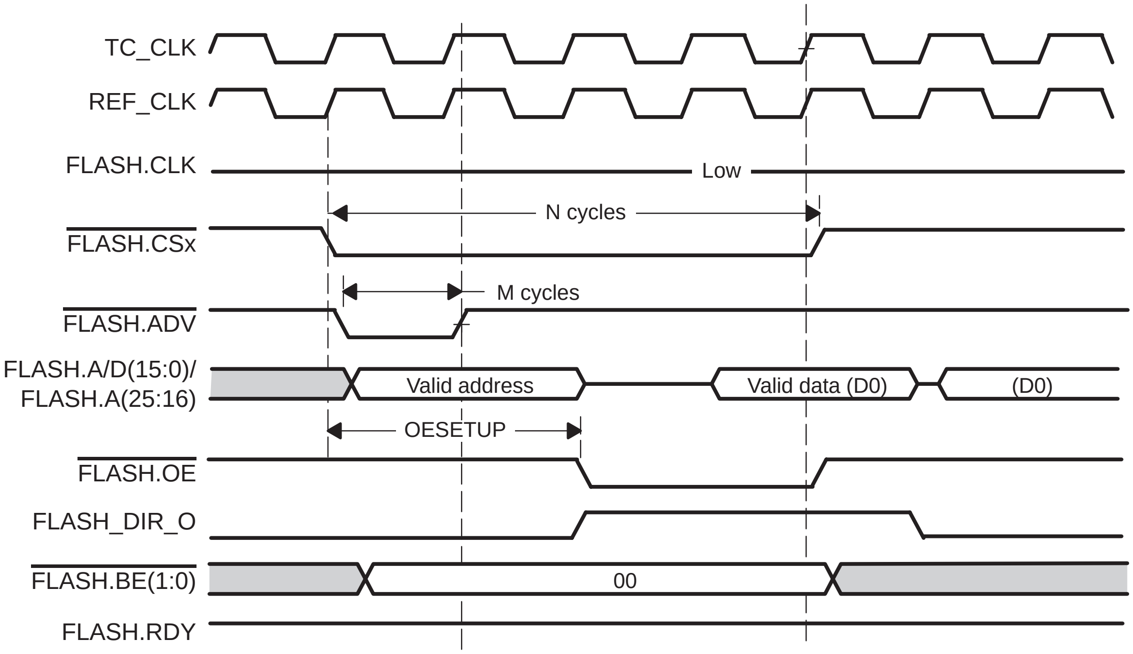
Figure 9. Asynchronous 16-Bit Read Operation with Ready. RDWST=2 FCLKDIV=0 OESETUP = 0 OEHOLD = 0 ADVHOLD = 0. Data write-back on the bus after read completion.



Asynchronous Read With Multiplexed Address and Data Memory

- ❑ The EMIFS can support multiplexed address and data memory devices without adding external logic. Multiplexed mode is enabled when the MAD bit field in the EMIFS CS configuration register is set to 1 (see Table 19).
- ❑ The following figure shows an asynchronous read operation with multiplexed address/data bus.

Figure 10. Asynchronous 16-Bit Read Operation With Multiplexed Address/Data Bus Memory. RDWST=2 FCLKDIV=0 OESETUP=2 OEHOLD=0 ADVHOLD=0. Data write-back on the bus after read completion.

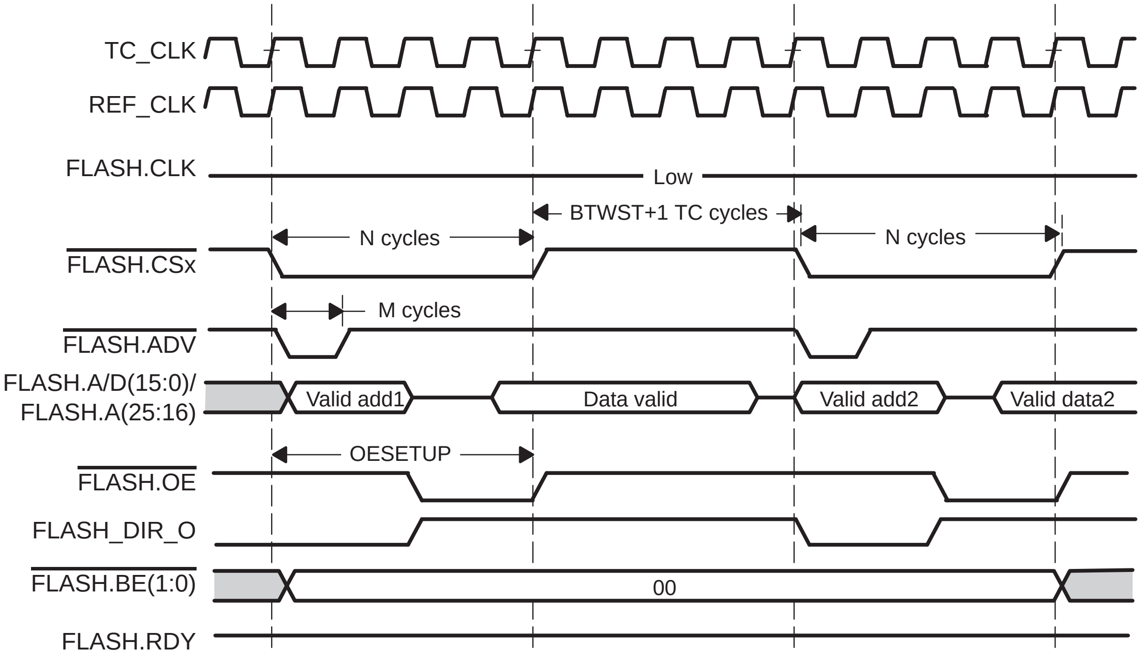


❑ Address drive time follows CS activation (no setup time guaranty). Address setup time to ADV rising edge is controlled by ADVHOLD. Address hold from ADV rising edge is guaranteed to be minimum one REF_CLK (delay for direction to change from out to in). FCLKDIV and OESETUP must be properly programmed to prevent bus contention and to ensure that the address hold time device requirement is respected.

❑ During split read accesses and during burst read accesses, the CS signal is deactivated for at least one TC_CK between two successive accesses. CS pulse width high time can be extended by the BTWST field in the CS configuration register (see also bus turn around and CS negation time control).

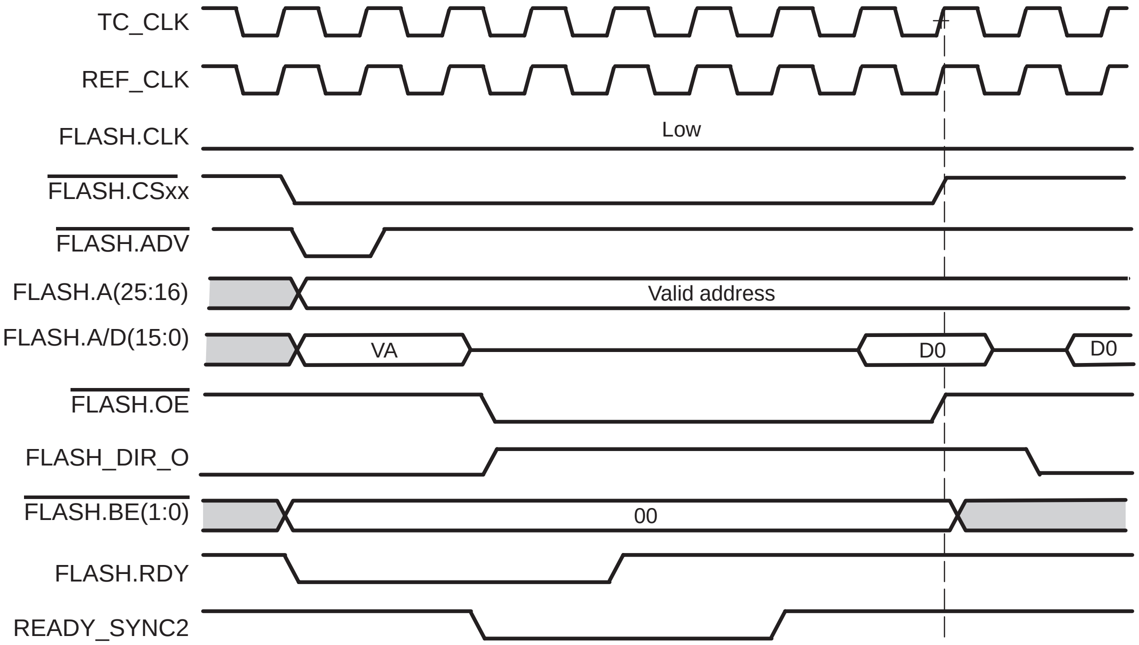
■ CS pulse width high = (BTWST +1) TC_CK

Figure 11. Asynchronous 32-Bit Read Operation on a 16-Bit Multiplexed Address and Data Memory. RDWST=2 FCLKDIV=0 OESETUP=2 OEHOLD = 0 ADVHOLD = 0



- The full-handshaking scheme is also valid in multiplexing mode. The following diagram shows an asynchronous word16 read operation with a 16-bit multiplexed memory control by external ready pin.

Figure 12. Asynchronous 16-Bit Read Operation with Ready on 16-Bit Multiplexed Address and Data Memory. RDWST=2 FCLKDIV=0 OESETUP=2 OEHOLD = 0 ADVHOLD = 0 BTWST = 0, BTMODE = 0

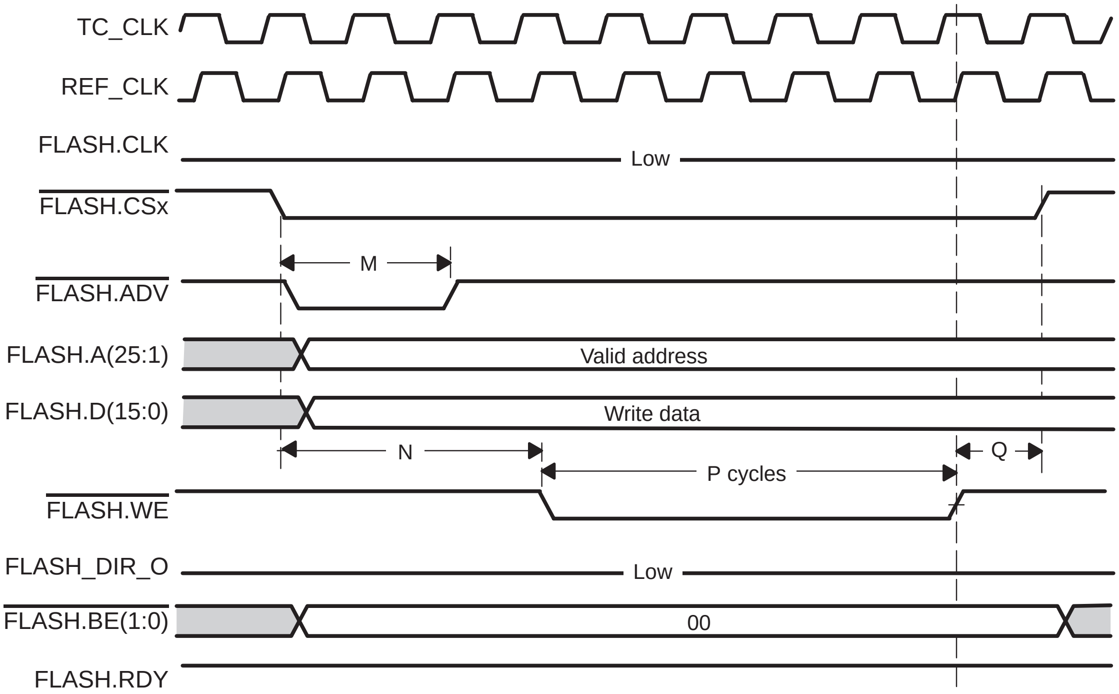


3.2.9 Asynchronous Write Operation

Non-Multiplexed Asynchronous Write Operation

❑ The asynchronous write is the only write protocol supported by the EMIFS. The asynchronous write access protocol is used in all EMIFS modes, whatever the CS configuration register RDMODE value is.

Figure 13. Asynchronous 16-Bit Write Operation on a 16-Bit Width Device (WRWST=2, WELEN=4 FCLKDIV=00 and ADVHOLD=1)



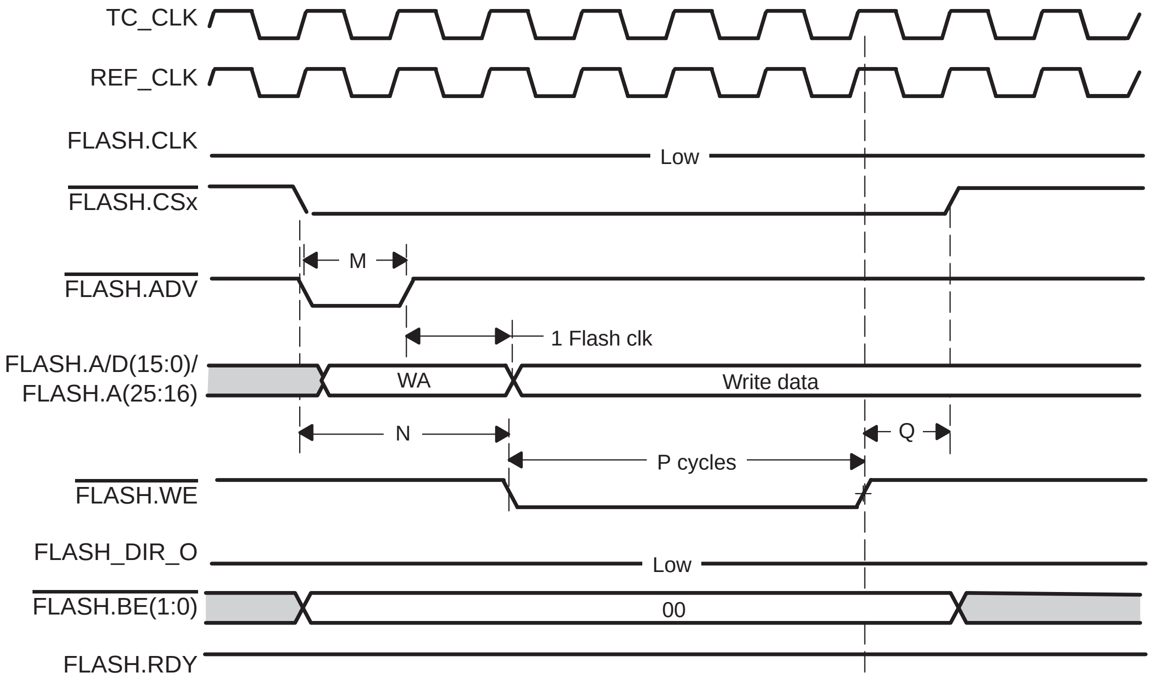
- ❑ The REF_CLK is divided from TC_CK by a programmable value contained in FCLKDIV bit field of the CS configuration register.
- ❑ The CS and address setup time from WE low is controlled by the programmable WRWST bit field of the CS configuration register.
  - ■ (WRWST + 1) REF_CLK (N cycles in Figure 13).
  - ■ WRWST minimum pulse width is 1 REF_CLK.
- ❑ The ADV pulse width depends on the ADVHOLD bit field of the Advanced CS configuration register (see Table 29). ADV pulse width equals:
  - ■ (ADVHOLD + 1) REF_CLK (M cycles in Figure 13).
  - ■ ADV minimum pulse width is 1 REF_CLK.
- ❑ The WE pulse width depends on the WELEN bit field of the CS configuration register (see Table 19). WE pulse width equals:
  - ■ (WELEN + 1) REF_CLK (P cycles in Figure 13).
  - ■ WE minimum pulse width is 1 REF_CLK.

- ❑ The CS address and data hold time setup from WE high is fixed to one REF_CLK (Q cycle in Figure 13).
- ❑ In asynchronous mode 0−1−2−3, REF_CLK is not provided outside the EMIFS and FLASH.CLK is kept low. In synchronous mode 4−5, REF_CLK is provided outside the EMIFS through the FLASH.CLK (see mode 4,5). In synchronous mode 7, REF_CLK is inverted and provided outside the EMIFS through FLASH.CLK (see mode 7).

Multiplexed Asynchronous Write Operation

Figure 14 shows a timing diagram with multiplexed address/data bus.

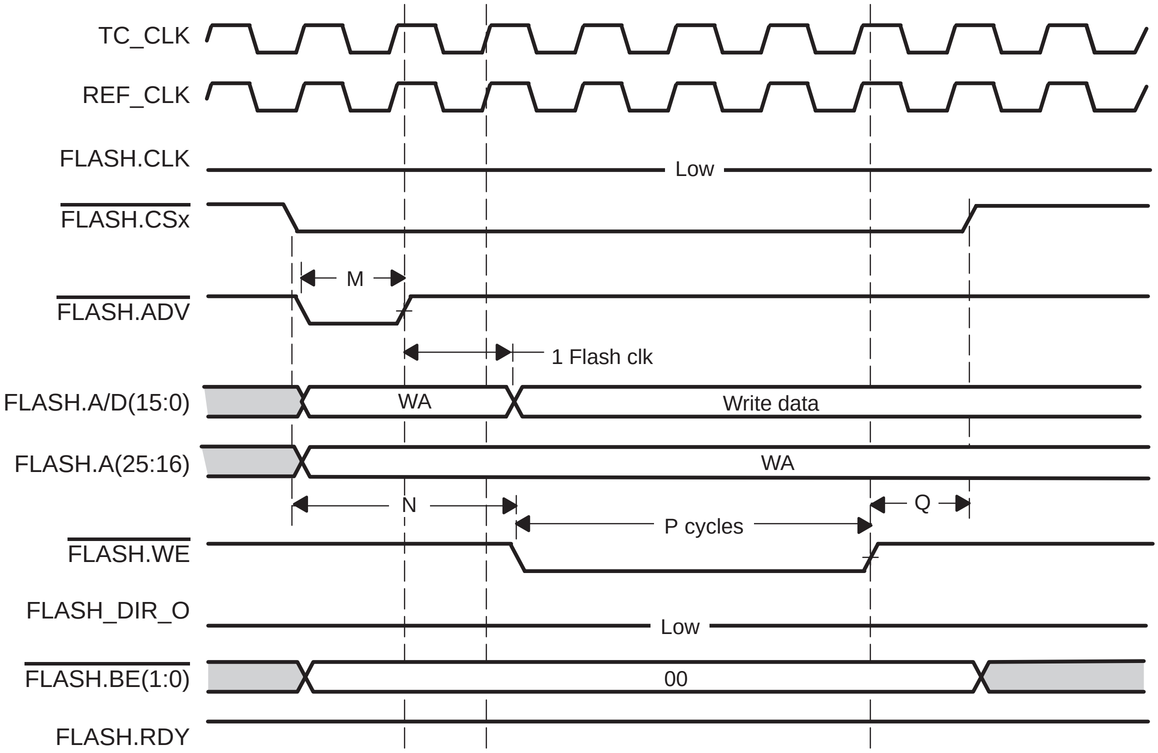
Figure 14. Asynchronous 16-Bit Write Operation on a Multiplexed Address/16-Bit Data Bus (WRWST=1, WELEN=3 , FCLKDIV=00 and ADVHOLD=0)



- ❑ Multiplexed mode is enabled when the MAD bit field in CS configuration register is set (see Table 19).
- ❑ Address drive time follows CS activation (no setup time guaranty). Address setup time to the ADV rising edge is controlled by ADVHOLD. Address invalid from the ADV rising edge is guaranteed to be minimum one REF_CLK cycle.

- The CS and address setup time from WE low is controlled by the programmable WRWST bit field of the CS configuration register (see Table 19).
  - (WRWST + 1) REF_CLK (N cycles in Figure 14).
  - WRWST minimum pulse width is 1 REF_CLK.
- The ADV pulse width depends on the ADVHOLD bit field of the advanced CS configuration register (see Table 29). ADV pulse width equals:
  - (ADVHOLD + 1) REF_CLK (Mcycles in Figure 14).
  - ADV min pulse width is 1 REF_CLK.
- The WE pulse width depends on the WELEN bit field of the CS configuration register (see Table 19). WE pulse width equals:
  - (WELEN + 1) REF_CLK (P cycles in Figure 14).
  - WE minimum pulse width is 1 REF_CLK.
- The CS and data hold time setup from WE high is fixed to one REF_CLK (Q cycle in Figure 14).
- In asynchronous mode 0−1−2−3, REF_CLK is not provided outside the EMIFS and FLASH.CLK is kept low. In synchronous mode 4−5, REF_CLK is provided outside the EMIFS through FLASH.CLK. In synchronous mode 7, REF_CLK is inverted and provided outside the EMIFS through the FLASH.CLK (see mode 7).

Figure 15. Asynchronous 16-Bit Write Operation on a Multiplexed Address/16-Bit Data Bus (WRWST = 1, WELEN = 3, FCLKDIV = 00 and ADVHOLD = 0)



Full-Handshaking and Ready Pin Usage in Asynchronous Write Mode

- ❑ In full-handshaking mode, FLASH.RDY is monitored by the EMIFS to control read access time. The access is completed when both internal wait state WELEN expires and FLASH.RDY (ball V2) is asserted by the external device. Full-handshaking is the default mode in both multiplexed and non-multiplexed modes.

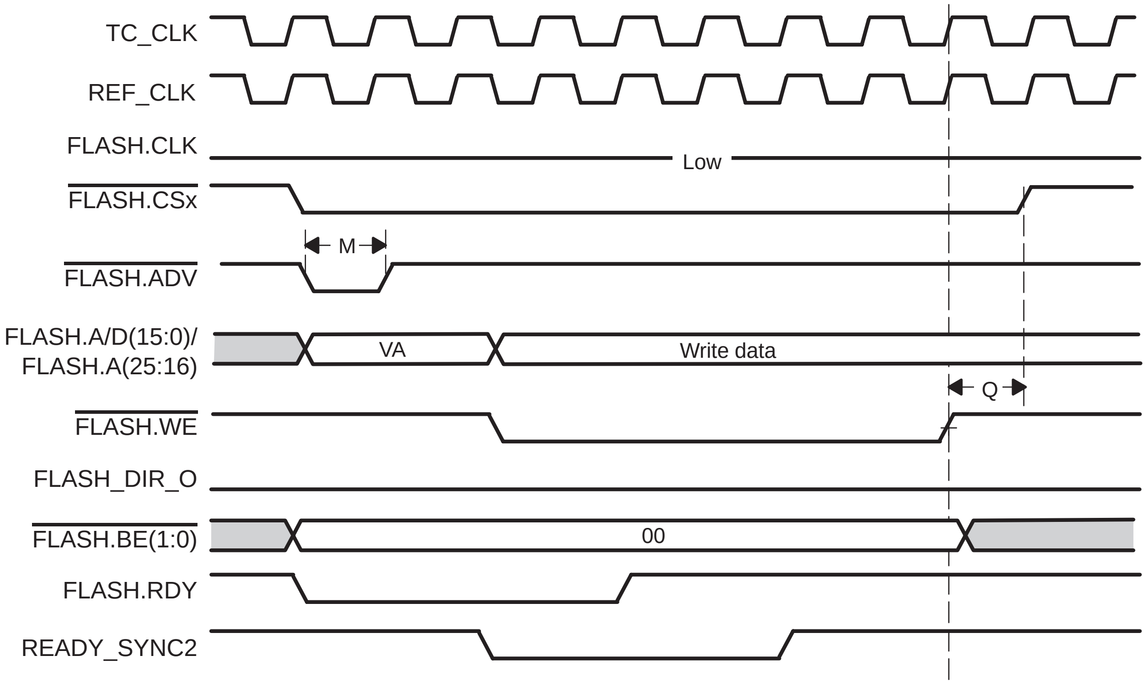
- ❑ ADV pulse width time and WE assertion time are not dependent on the FLASH.RDY state.

- ❑ When FLASH.RDY is used to extend the access time, the access completion is not controlled by internal delay generation. The CS and data hold time setup from WE high is still fixed to one REF_CLK (Q cycle in previous figures).

❑ Because FLASH.RDY is an asynchronous signal, a nonready device must drive FLASH.RDY low enough time ahead of the minimum access time completion. Depending on FLASH.RDY assertion low delay from CS active and depending on REF_CLK frequency, a minimum WRWST value could be needed for the EMIFS to correctly monitor the FLASH.RDY state.

- As an example, a minimum of WRWST=1 is needed for a nonready device that drives ready low with 0 time delay from CS low, for a CS configuration FCLKDIV=0.

Figure 16. Asynchronous 16-Bit Write Operation on 16-Bit Multiplexed Address and Data Memory With Ready (WELEN = 2, WRWST = 0, FCLKDIV = 0)

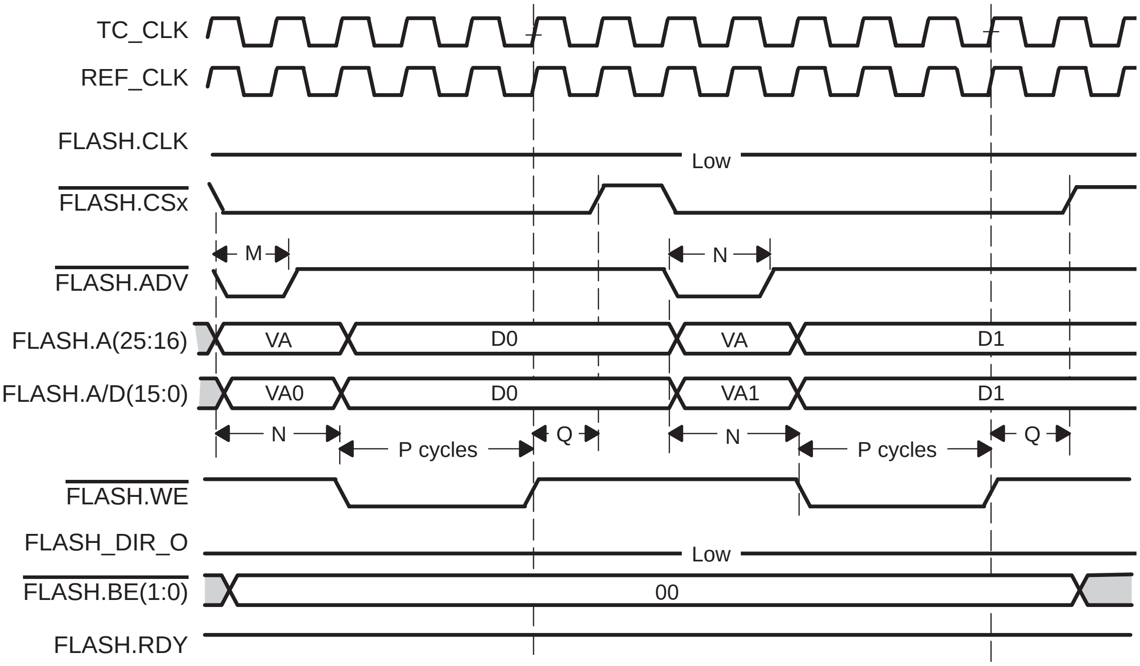


Write Access Size Adaptation and CS Pulse Width High Control

❑ During write access, the EMIFS splits the Word32 access into two Word16 accesses in case of 16-bit device width. 4xWord32 burst writes are split into 8 successive Word16 accesses. The split process follows the little endian protocol (Word32 LSB part at lower Word16 address).

- ❑ During split write accesses and during burst write accesses, the CS signal is not deactivated unless BTMODE in the Advanced CS configuration register is set (see Table 29). When BTMODE is set, the CS pulse width high time can be controlled by the BTWST field in the CS configuration register, Table 19 (see also bus turn around and CS negation time control).
  - ■ CS pulse width high= (BTWST +1) TC_CK
- ❑ This applicable to both multiplexed and non-multiplexed access modes.

Figure 17. Asynchronous 32-Bit Write Operation on 16-Bit Multiplexed Address and Data Memory (WELEN = 2, WRWST =1, FCLKDIV =0, BTWST = 0, and BTMODE = 1)



3.2.10 Mode 1−2−3− Asynchronous Page Mode Read Operation

- ❑ The asynchronous page mode read 1−2−3 is selected by setting the RDMODE bit field in the corresponding EMIFS chip-select configuration register.
  - ■ RDMODE = 1 selects the 4 words per page mode.
  - ■ RDMODE = 2 selects the 8 words per page mode.
  - ■ RDMODE = 3 selects the 16 words per page mode.

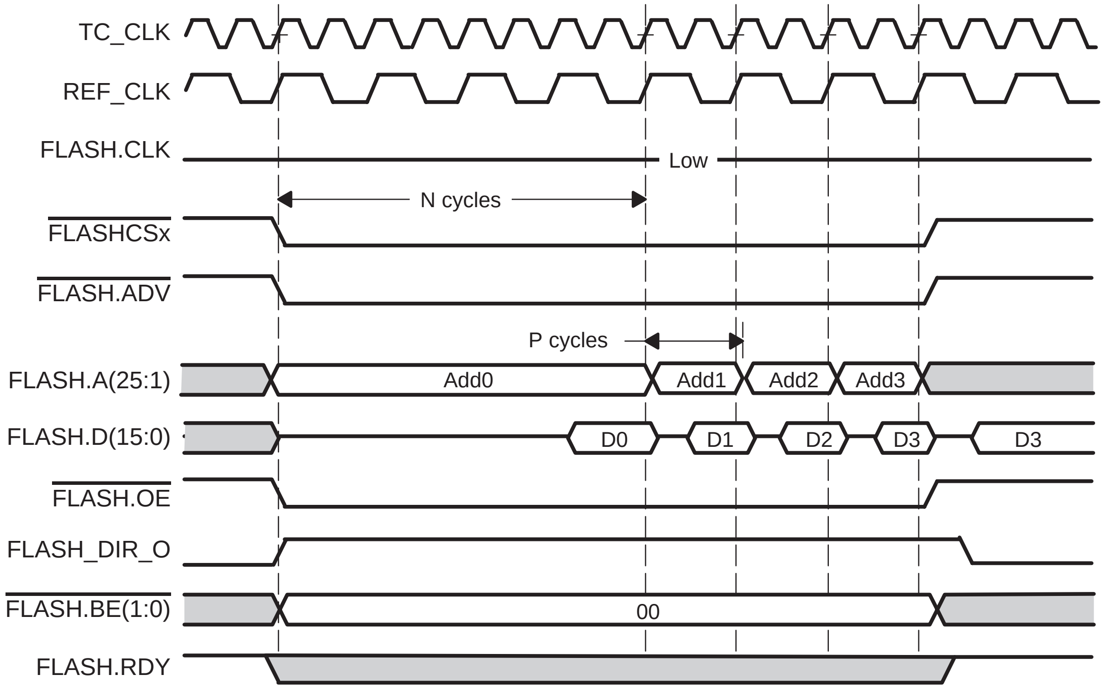
❑ This mode provides single access or fast consecutive accesses in a page. Access time is optimized for 2×word16 read (word32 read in 16-bit width device) and for burst read (4×word32 or 8×Word16 in case of 16-bit width device). During consecutive accesses, the EMIFS increments the address after each word read completion.

❑ The word length of the access is equal to the memory data bus width and is defined by the BW field of the CS configuration register (see Table 19).

❑ The delay for the first word in the page is controlled by RDWST bit field in the CS configuration register (initial wait state). Depending on the device page length and word size (device width), the EMIFS can control device page crossing during a burst request (4×word32) and can insert initial wait state delay on purpose.

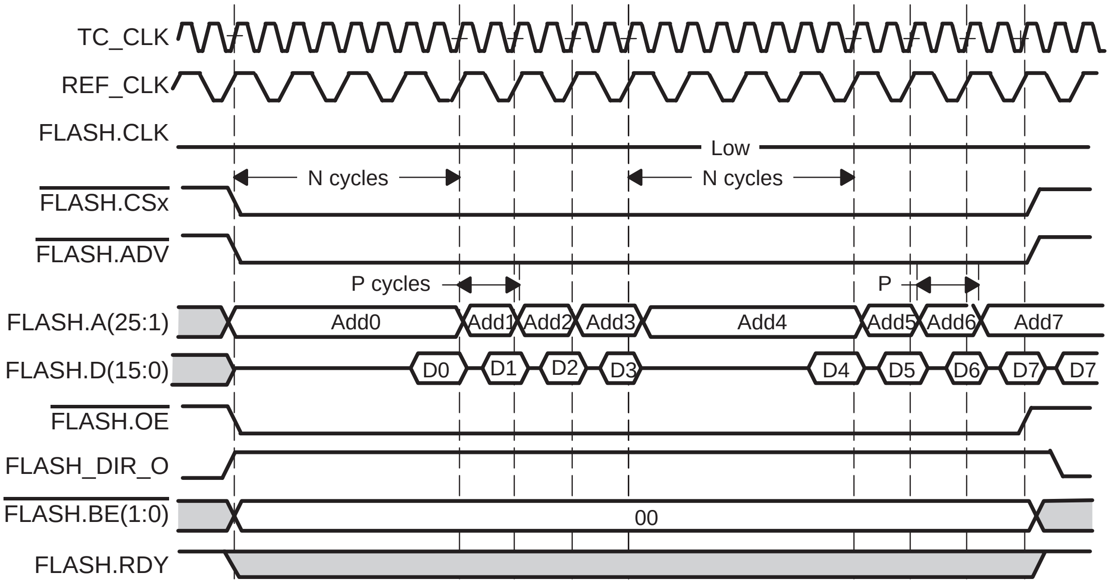
❑ The delay between successive words in the page is controlled by the PGWST bit field in the CS configuration register (in page wait state).

Figure 18. Asynchronous Page Mode 4x16-Bit Read Operation on 16-Bit Width Device (RDWST=2, PGWST=0 and FCLKDIV =1, RDMODE=2). Data write-back on the bus after read completion.



- The REF_CLK is divided from TC_CK by a programmable value contained in FCLKDIV bit field of the CS configuration register (Table 19).
- The initial wait state depends on RDWST bit field of the CS configuration register. Delay equals:
  - (RDWST + 2) REF_CLK (N cycles in Figure 18).
- The in page wait state depends on:
  - PGWST/WELEN[15:12] bit field, if PGWSTEN=0 in CS configuration register.
  - PGWST[30:27] bit field, if PGWSTEN=1 in CS configuration register.
  - Delay equals (PGWST + 1) REF_CLK (P cycles in Figure 18).
- ADV ($\overline{\text{FLASH.ADV}}$ on ball L4) is kept low for the entire access.
- Address drive time follows CS activation (no setup time guaranty).
- Delay time (OESETUP) and advanced time (OEHOLD) are disabled (OESETUP and OEHOLD bit fields don’t care).
- Address and data multiplexed scheme is not supported in mode 1−2−3.
- Read data are latched on the TC_CK rising edge corresponding to the wait state delay completion (initial and in page wait state).
- One TC_CK cycle after access completion (CS high), the data bus is driven with the previous read value (see Figure 18 for direction activation and data copy timing).
- In asynchronous mode, REF_CLK is not provided outside the EMIFS and FLASH.CLK is kept low.
- Page mode always follows the non-full-handshaking protocol and the FLASH.RDY pin is never monitored whatever the full-handshaking bit field value in the dynamic wait state register is.

Figure 19. Asynchronous Page Mode 8x16-Bit Read With Page Crossing Operation on 16-Bit Width Device (RDWST=2, PGWST=0 FCLKDIV=1, RDMODE=1). Data write-back on the bus after read completion.

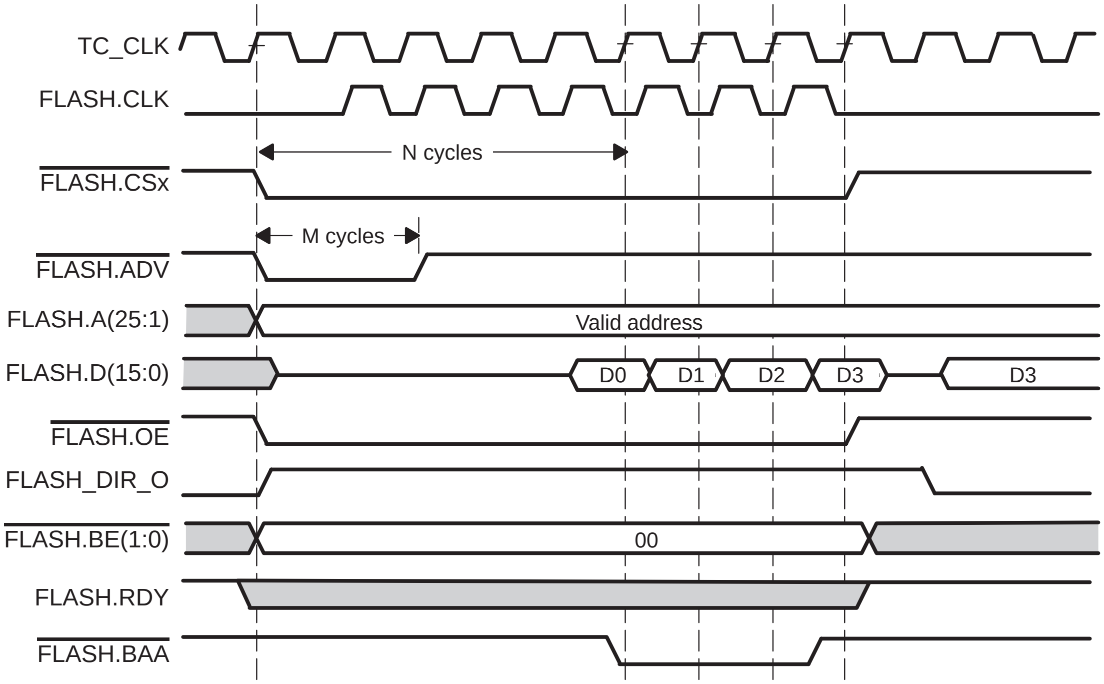


3.2.11 Mode 4 and Mode 5 Synchronous Burst Read Operation Mode

Synchronous Read in Non-Multiplexed Address and Data Memory

- The synchronous burst read mode 4 and 5 are selected by setting the RDMODE bit field in the corresponding EMIFS chip-select configuration register (see Table 19).
  - RDMODE = 4 or 5.
- This mode only supports synchronous read accesses (single or consecutive). Flash devices usually require synchronous setup and enable mode to be done after power up. RDMODE must be changed to mode 4 or mode 5 only after flash device setup.

Figure 20. Mode 4 Synchronous Burst 4x16-Bit Read Operation on 16-Bit Width Device (RDWST=3, FCLKDIV =0, ADVHOLD=0, RDMODE=4). Data write-back on the bus after read completion.



- ❑ The REF_CLK is divided from TC_CK by a programmable value contained in FCLKDIV bit field of the CS configuration register (see Table 19).
- ❑ CS, ADV and address are driven one REF_CLK cycle before the first FLASH.CLK rising edge is provided externally. This ensures that CS, ADV, and address valid setup time to device clock rising edge will be met.
- ❑ In case this one REF_CLK cycle advance is not enough to meet the setup time requirement, the ADV pulse width can be extended by ADVHOLD. The real access time starts from CS, ADV, and address setup time to device clock rising edge valid.
- ❑ The ADV pulse width depends on the ADVHOLD bit field of the Advanced CS configuration register (see Table 29). ADV pulse width equals:
  - ■ (ADVHOLD + 1) REF_CLK + 1 TC_CK (M cycles in Figure 20).
- ❑ Modes 4−5 are by default in full-handshaking mode. FLASH.RDY is monitored by the EMIFS to control read access time. FLASH.RDY must be asserted synchronously to FLASH.CLK.

- ❑ The first access is completed when both the internal RDWST wait state has expired and FLASH.RDY is asserted by the external device.
- ❑ The internal initial wait state depends on RDWST bit field of the CS configuration register. RWDST value must include the extra non-active output REF_CLK cycle used for CS, ADV, and address setup time. Delay equals:
  - ■ (RDWST + 2) REF_CLK (N cycles in Figure 20).
- ❑ Read data are latched on each TC_CK rising edge corresponding to a REF_CLK rising edge when FLASH.RDY has been sampled high on the previous REF_CLK rising edge.
- ❑ The following in-burst access wait state only depends on the FLASH.RDY state (RDWST expired).
- ❑ In mode 4, BAA control signal is asserted low on the first date sampling REF_CLK rising edge and is maintained low during the full burst access. BAA is kept high in mode 5 (no burst advance control in this mode).
- ❑ OE activation delay time from CS and address valid is programmable through the OESETUP bit field in the advanced CS configuration register. Activation delay timing is equal to:
  - ■ (OESETUP) REF_CLK.
- ❑ Advanced time (OEHOLD) control is disabled (OEHOLD bit field doesn’t care).
- ❑ One TC_CK cycle after access completion (CS high), the data bus is driven with the previous read value (see Figure 20 for direction activation and data copy timing).

Figure 21. Mode 5 Synchronous Burst 8x16-Bit Read Operation on 16-Bit Width Device (RDWST=3, FCLKDIV =0, ADVHOLD=0, RDMODE=5). Data write-back on the bus after read completion.

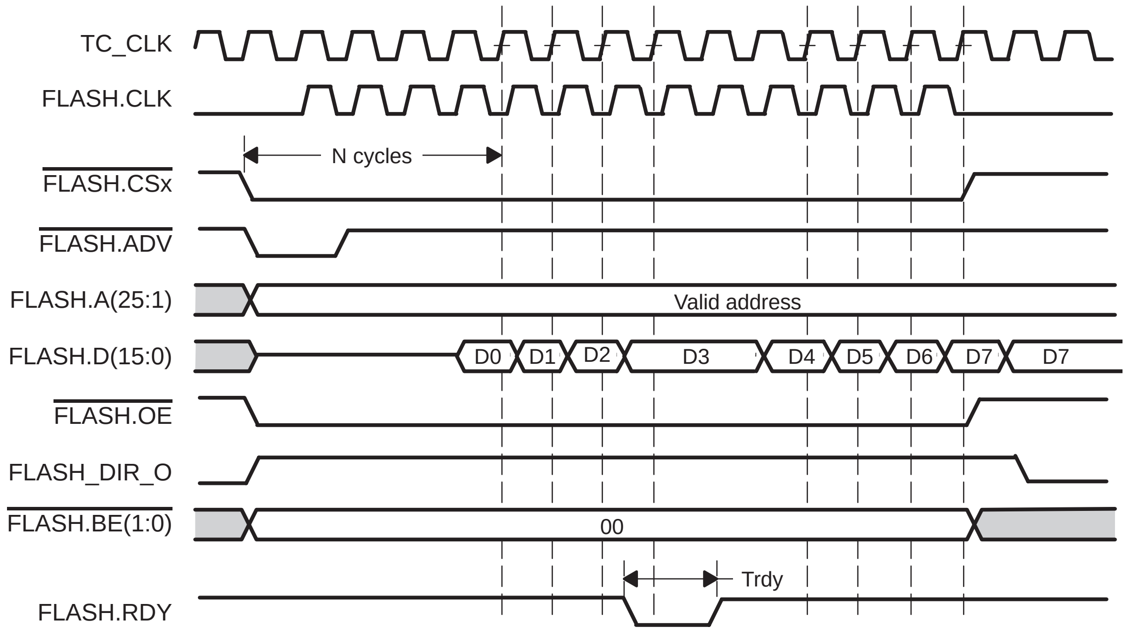
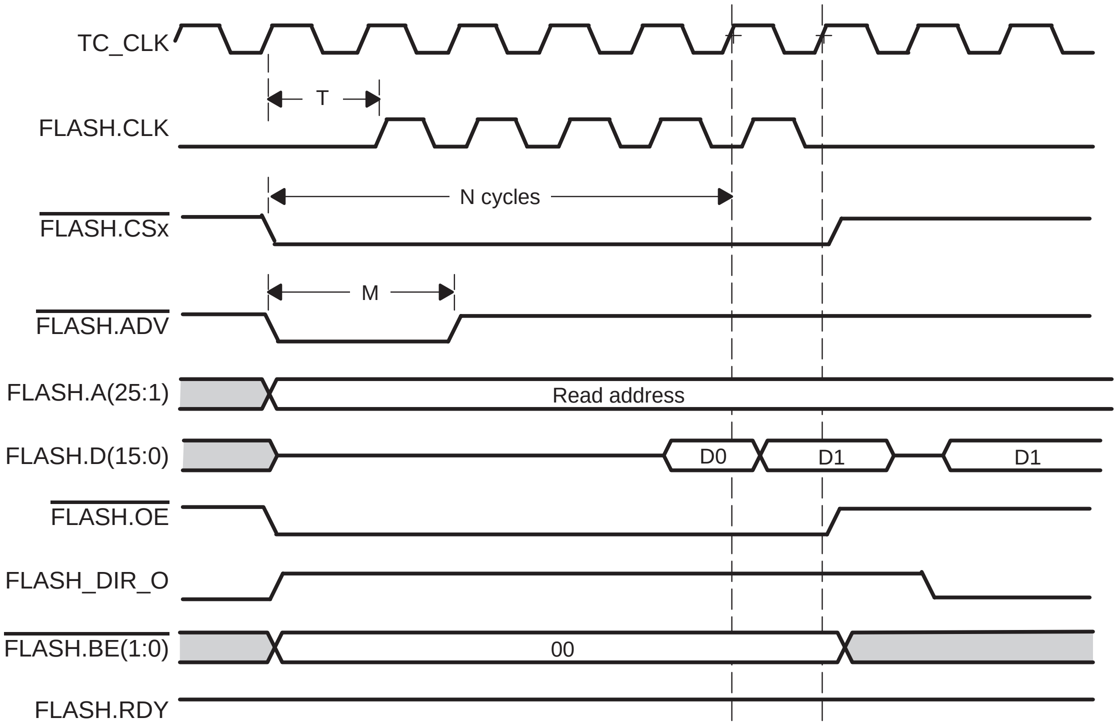


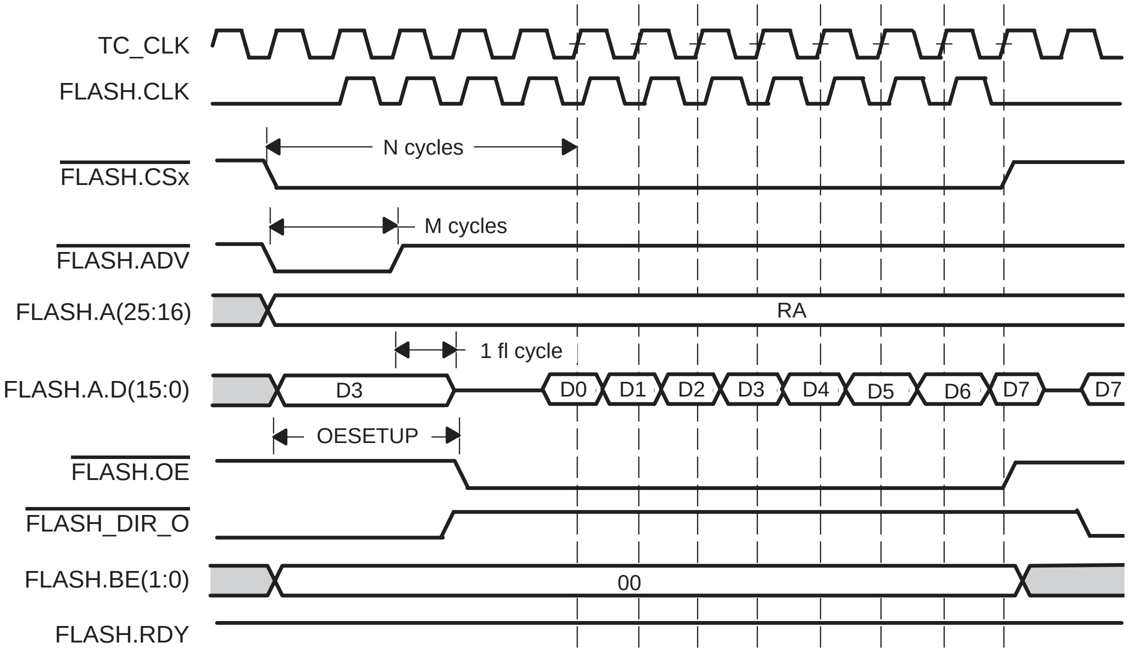
Figure 22. Mode 5 Synchronous Burst 2x16-Bit Read Operation on 16-Bit Width Device (RDWST=3, FCLKDIV =0, ADVHOLD=0, RDMODE=5). Data write-back on the bus after read completion



Synchronous Read in Multiplexed Address and Data Memory

❑ Multiplexed mode is enabled when the MAD bit field in CS configuration register is set to 1 (see Table 19).

Figure 23. Mode 5 Synchronous Burst 8x16-Bit Read Operation on Multiplexed Address/Data 16-Bit Width Device (RDWST=2, FCLKDIV =0, ADVHOLD=0, OESETUP = 3, RDMODE=5). Data write-back on the bus after read completion.



- ❑ CS, ADV, and address are driven one REF_CLK cycle before the first FLASH.CLK rising edge is provided externally. This ensures CS, ADV, and address valid setup time to device clock rising edge to be met.
- ❑ In case this one REF_CLK cycle advance is not enough to meet the setup time requirement, the ADV pulse width can be extended by ADVHOLD. The real access time start from CS, ADV, and address setup time to device clock rising edge valid.
- ❑ Address hold time from ADV rising edge is guaranteed to be a minimum of one REF_CLK (delay for direction to change from out to in).
- ❑ FCLKDIV and OESETUP (REF_CLK) must be properly programmed to prevent bus contention and to ensure that address hold time device requirement is respected.
- ❑ Delay time OEHOLD is disabled.
- ❑ The ADV pulse width depends on ADVHOLD bit field of the Advanced CS configuration register (see Table 29). ADV pulse width equals:
  - ■ (ADVHOLD + 1) REF_CLK + 1 TC_CK (M cycles in Figure 23)

- ❑ Modes 4−5 are in full-handshaking mode by default. FLASH.RDY is monitored by the EMIFS to control read access time. FLASH.RDY must be asserted synchronously to REF_CLK.
- ❑ The first access is completed when both internal RDWST wait states are expired and when the ready pin is asserted by the external device.
- ❑ The internal initial wait state depends on the RDWST bit field of the CS configuration register. RWDST value must include the extra non-active output REF_CLK cycle used for CS, ADV, and address setup time. Delay equals:
  - ■ (RDWST + 2) REF_CLK (N cycles in Figure 23)
- ❑ Read data is latched on each TC_CK rising edge corresponding to a REF_CLK rising edge when FLASH.RDY has been sampled high on the previous REF_CLK rising edge.
- ❑ The following in-burst access wait state only depends on the FLASH.RDY pin state (RDWST expired).
- ❑ One TC_CK cycle after access completion (CS high), the data bus is driven with the previous read value (see Figure 23 direction activation and data copy timing).

Figure 24. Mode 5 Synchronous Burst 4x16-Bit Read Operation on Multiplexed Address/Data 16-Bit Width Device (RDWST=4, FCLKDIV =0, ADVHOLD=1, OESETUP = 4, RDMODE=5). Data write-back on the bus after read completion.

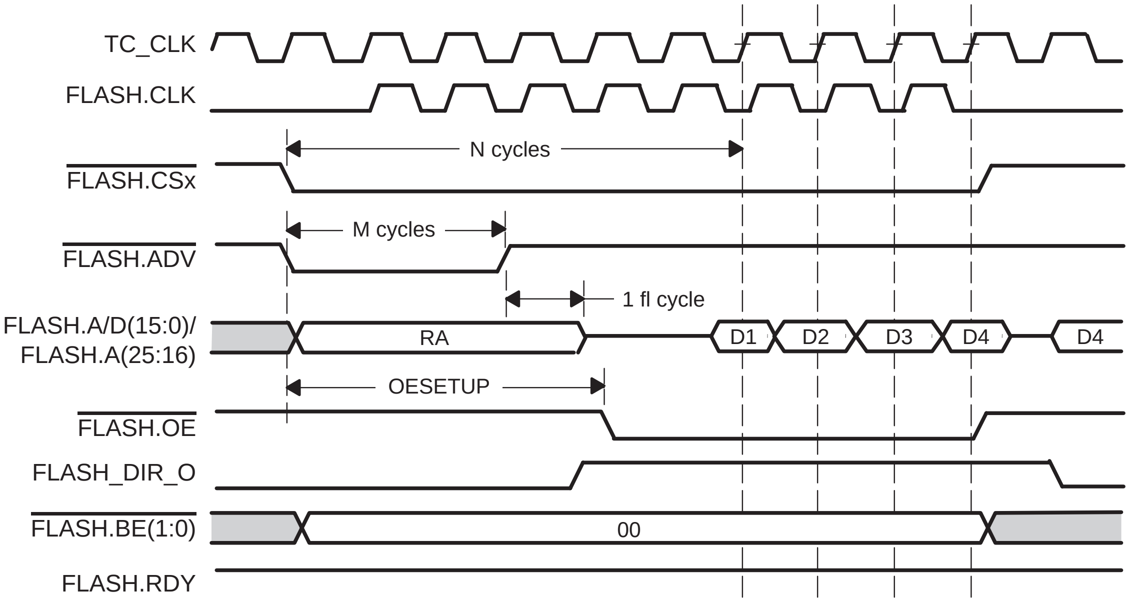
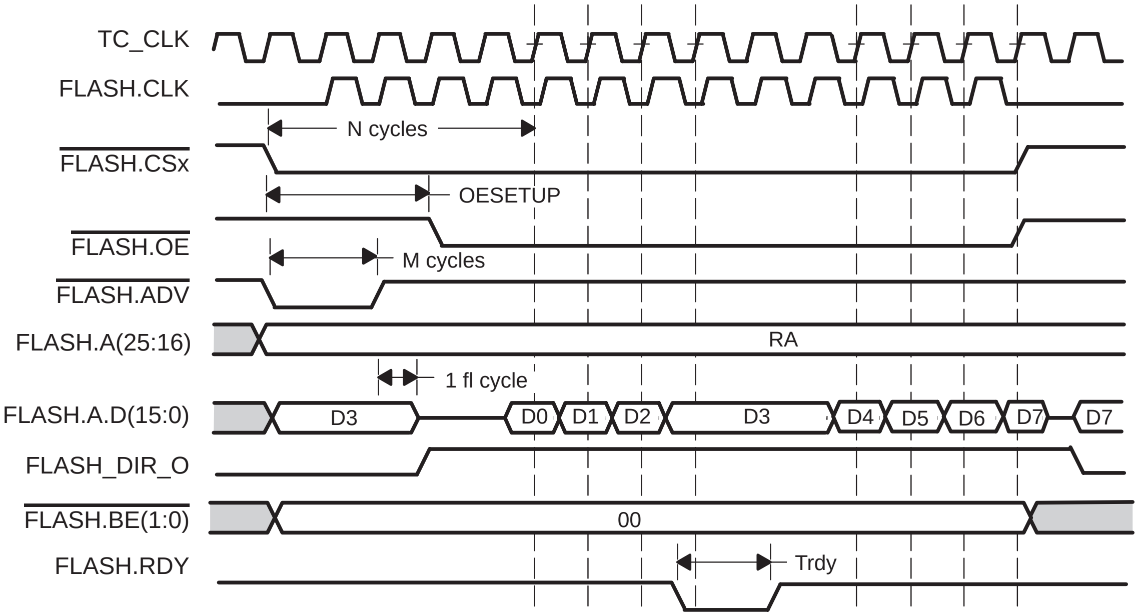


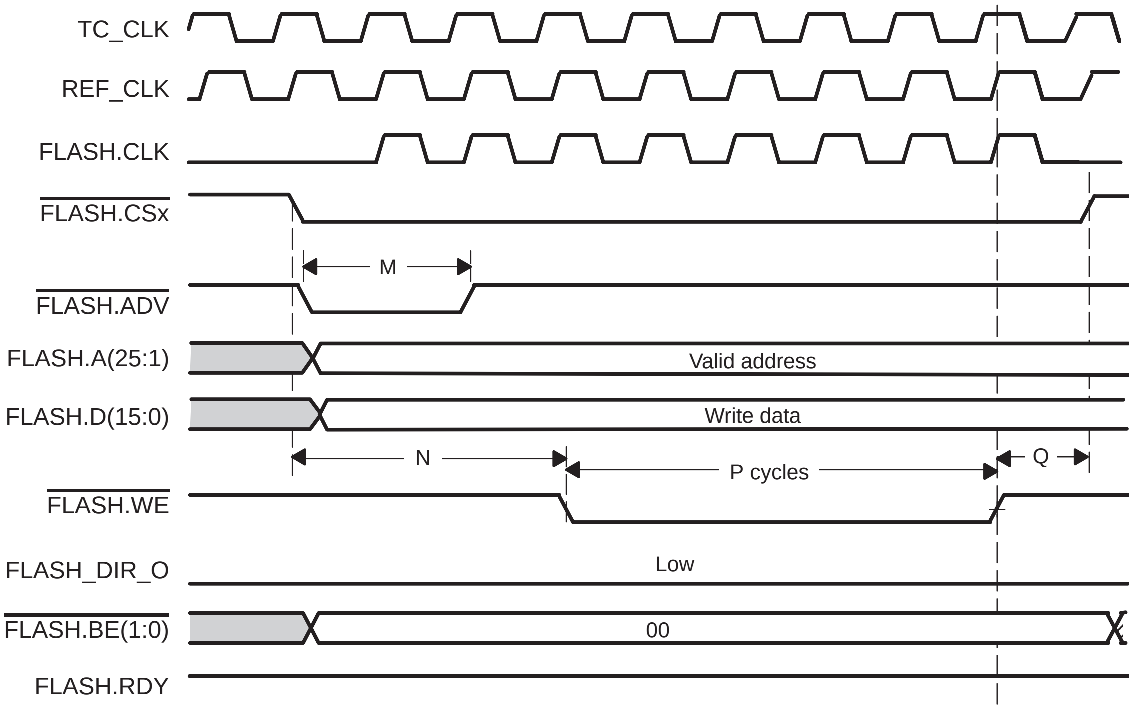
Figure 25. Mode 5 Synchronous Burst 8x16-Bit Read Operation on Multiplexed Address/Data 16-Bit Width Device (RDWST=3, FCLKDIV =0, ADVHOLD=0, OESETUP = 3, RDMODE=5). Data write-back on the bus after read completion.



Write Access in Mode 4 and 5

- ❑ Figure 26 shows FLASH.CLK activation details during a write access in mode 5 (non-multiplexed). Same behavior for the multiplexed address and data protocol.

Figure 26. Asynchronous 16-Bit Write Operation on a 16-Bit Width Device (RDMODE = 5, WRWST=2, WELEN=4 FCLKDIV=00 and ADVHOLD=1)



3.2.12 Read Retimed Protocol

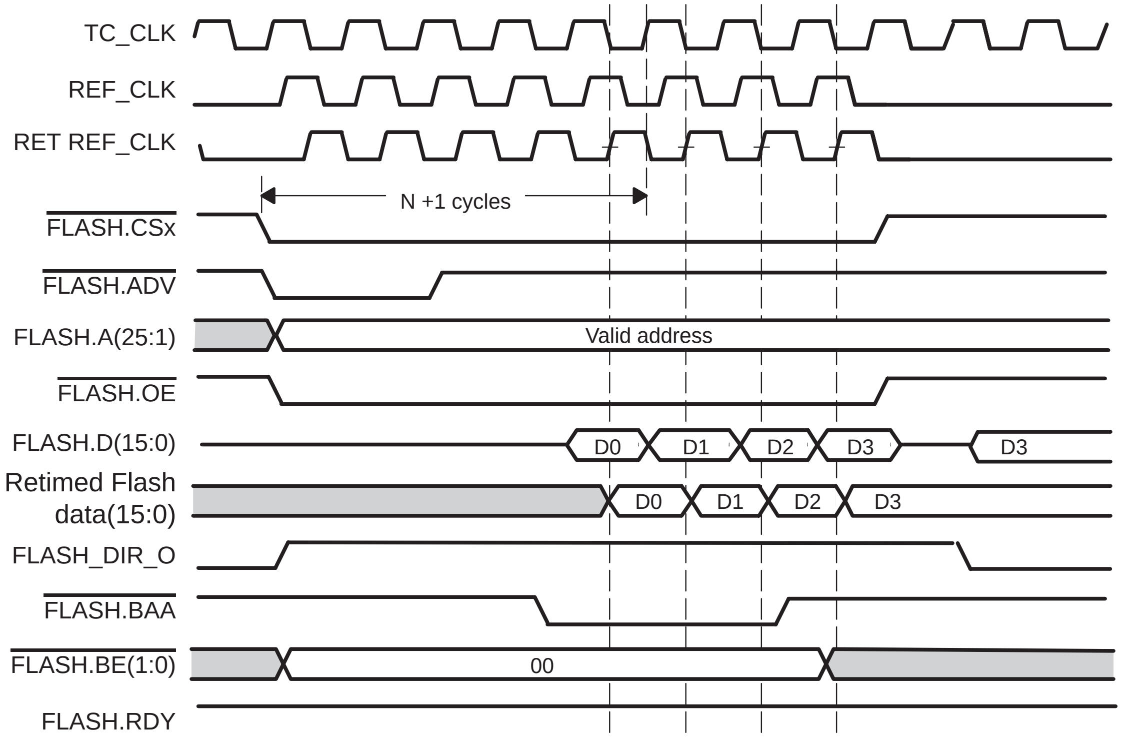
Warning

The RT bit in the EMIFS chip−select configuration register may be set only in RDMODE 4, 5, and 7 only. The system hangs if the retiming bit is set in other modes, such as asynchronous modes, because the retiming logic depends on the returned flash clock. In asynchronous mode, there is no flash clock and the system hangs.

- Due to IC I/O and board delays, the theoretical external memory maximum frequency may not be usable for REF_CLK value without retiming function. In synchronous mode 4-5, the retiming mode allows read data to be latched by a delayed Ret_REF_CLK obtained through the IC I/O feedback of FLASH.CLK. This offers optimum data and sampling clock alignment.

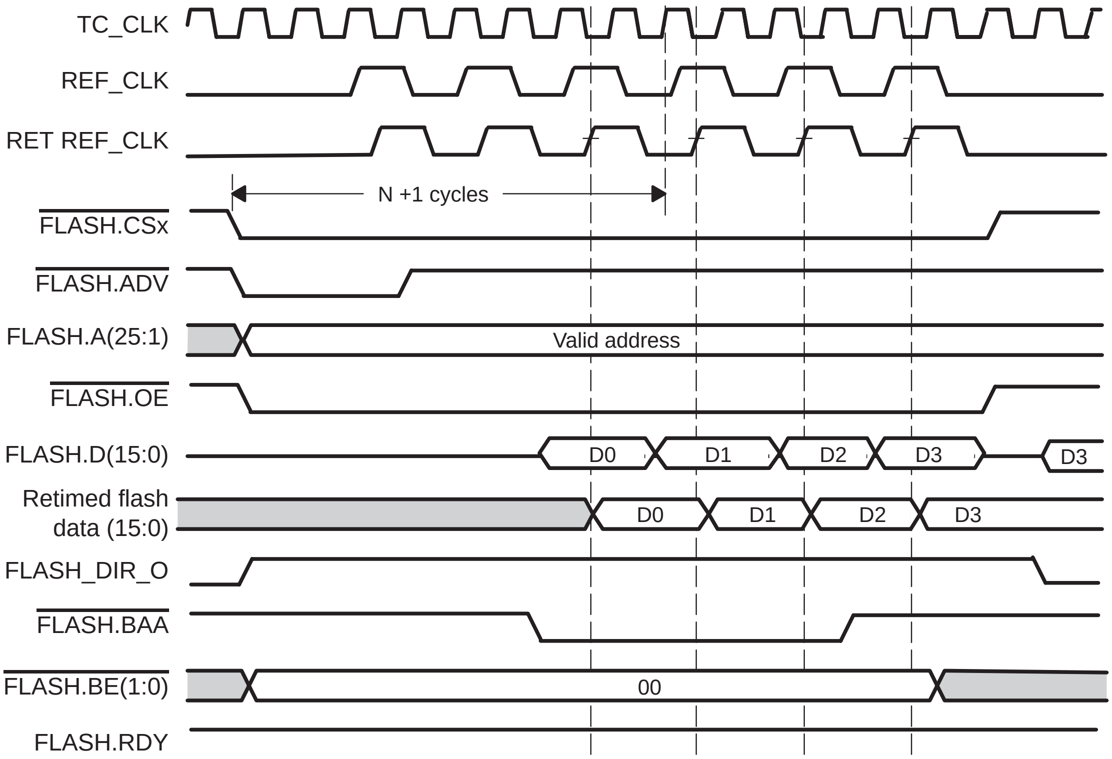
- ❑ Retiming mode enables a pipelined read access. Compared to non-retimed access, the first access takes one extra REF_CLK cycle and the following accesses in a burst take one REF_CLK cycle each.
- ❑ The retiming mode is enabled through the RT bit field in the CS configuration register. Retiming mode is only allowed in synchronous modes 4−5−7 and has no effect on write accesses.

Figure 27. Mode 4 Synchronous Burst 4x16-Bit Read Operation on 16-Bit Width Device With Retiming on (RDWST=2, FCLKDIV =0, ADVHOLD=0, RDMODE=4). Data write-back on the bus after read completion.



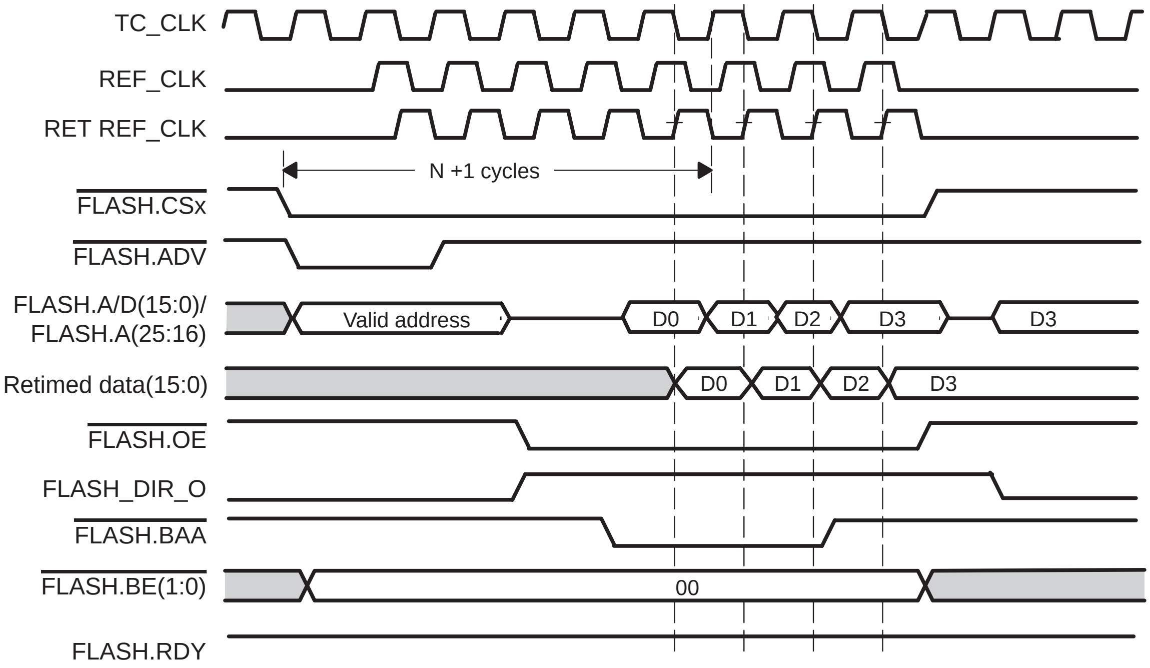
- ❑ As in non-retimed mode, CS, ADV, Address, BE, OE, and BAA are driven with respect to REF_CLK.
- ❑ Modes 4 and 5 programming model and protocol behavior remain the same as in non-retimed mode.
- ❑ In retiming mode, the RDWST is still referenced to REF_CLK. The retiming relaxed timing (extra delay for data valid) is included in the IC timing parameters.
- ❑ FLASH.RDY is also retimed with the Ret_REF_CLK. The retiming relaxed timing (extra delay for ready valid) is included in the IC timing parameters.

Figure 28. Mode 4 Synchronous Burst 4x16-Bit Read Operation on 16-Bit Width Device With Retiming on (RDWST=1, FCLKDIV =1, ADVHOLD=0, RDMODE=4)



- ❑ Retiming mode is also available in multiplexing address and data mode.

Figure 29. Mode 4 Synchronous Burst 4x16-Bit Read Operation on Multiplexed Address/Data 16-Bit Width Device With Retiming on (RDWST=3, FCLKDIV =0, ADVHOLD=0, OESETUP = 3, RMODE=4). Data write-back on the bus after read completion.

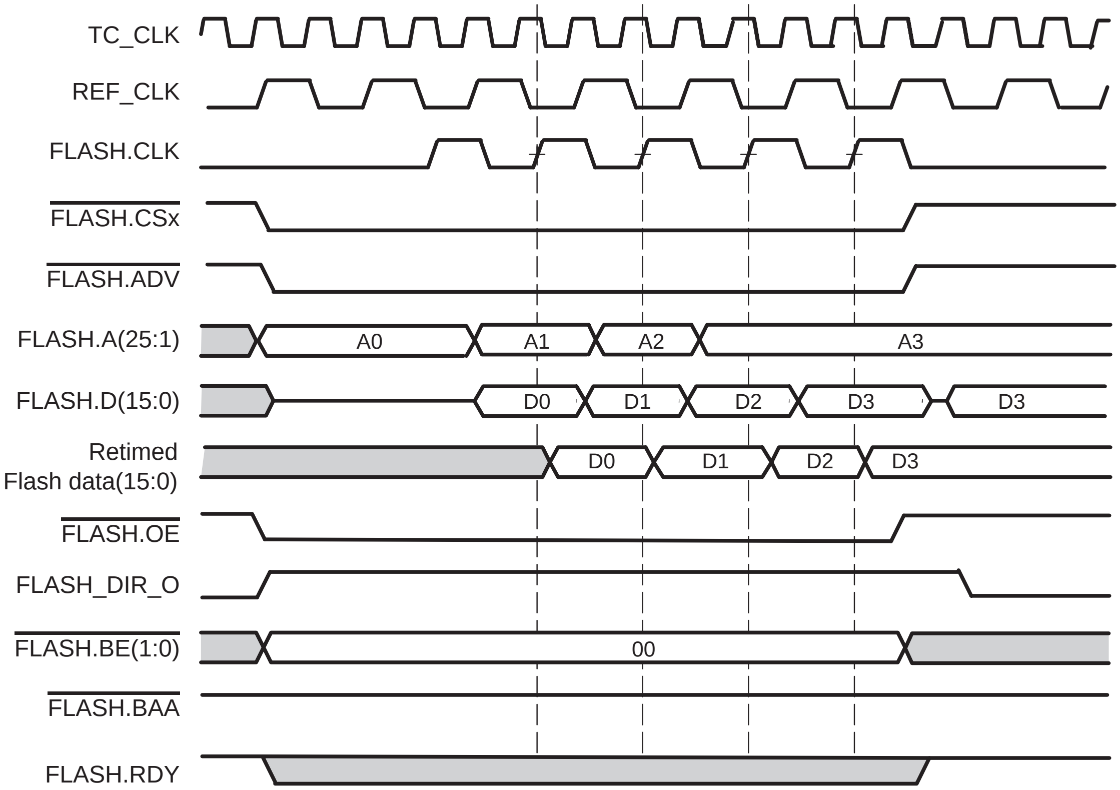


3.2.13 Mode 7—Synchronous Burst Read Operation Mode

- ❑ The synchronous burst read mode 7 is selected by setting the RDMODE bit field in the corresponding EMIFS chip-select configuration register.
  - ■ RDMODE = 7
- ❑ This mode only supports synchronous read accesses (single or consecutive). This protocol is used when accessing boot ROM.

The following diagrams show 16−bit external device widths. Similar accesses occur when using the 32−bit width of the boot ROM.

Figure 30. Mode 7 Synchronous Burst 4x16-Bit Read Operation on 16-Bit Width Device (RDMODE = 7, FCLKDIV =1). Data write-back on the bus after read completion.



- ❑ The REF_CLK is divided from TC_CK by a programmable value contained in the FCLKDIV bit field of the CS configuration register. REF_CLK is inverted and provided externally as FLASH.CLK.

- ❑ The retimed mode must be enabled in mode 7. After reset, the RT bit is set for CS0 and CS3 if mode 7 is selected during the reset boot configuration process (see Table 18).

- ❑ Mode 7 enables a pipelined read access. Within the burst, the addresses are provided one half FLASH.CLK cycle before the FLASH.CLK rising edge. Use the FLASH.CLK rising edge to latch addresses into the embedded ROM/RAM. The ROM/RAM must provide data with enough setup so data can be latched by the EMIFS on the next FLASH.CLK rising edge.

- ❑ REF_CLK frequency must be set so that the memory access time can be met in less than one cycle. There is no internal or external wait state control during read access in mode 7. The RDWST field is not active and FLASH.RDY is not monitored in this mode (non-full-handshaking mode only).
- ❑ CS, ADV, OE are driven low for the entire access. ADVHOLD, OESETUP, OEHOLD time control are disabled in this mode.
- ❑ Address and data multiplexed protocol is not supported in mode 7 (MAD bit field not considered).
- ❑ One TC_CK cycle after access completion (CS high) the data bus is driven with the previous read value.

Write Access in Mode 7

- ❑ Figure 31 shows FLASH.CLK activation details during a write access in mode 7. The example below shows two successive write accesses with minimum access time.

Figure 31. Mode 7 Asynchronous Two Successive 16-Bit Write Operations on a 16-Bit Width Device (WRWST=0, WELEN=0 FCLKDIV=1 and ADVHOLD=0)

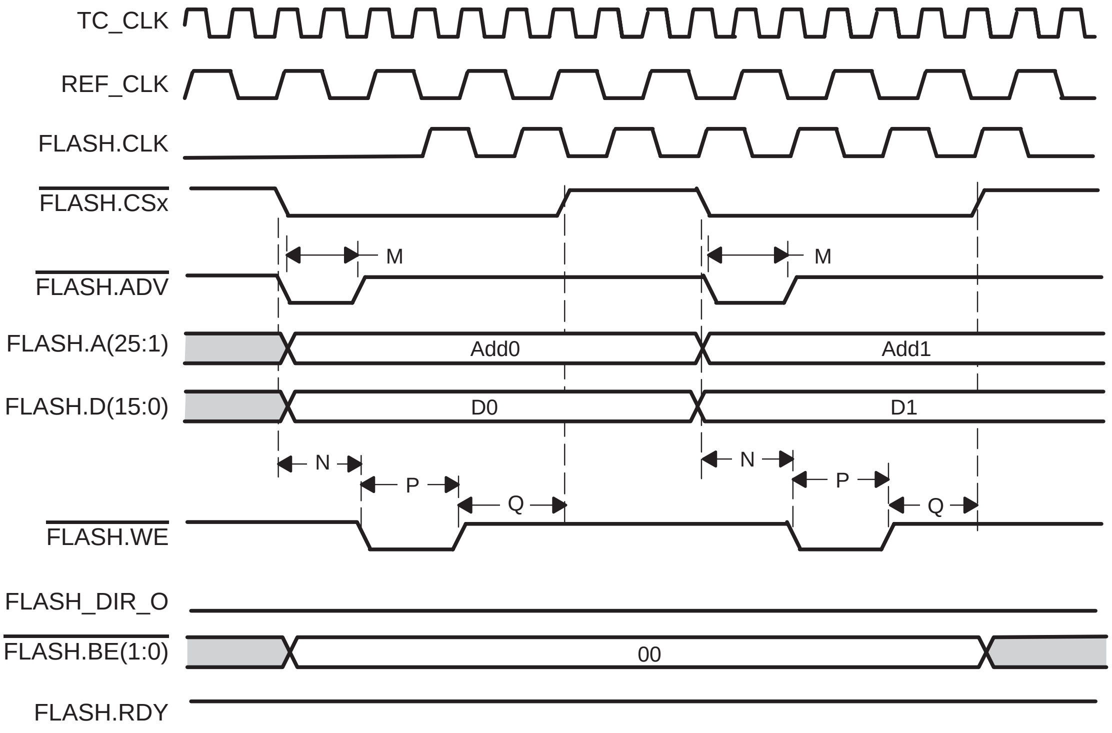


Figure 32. Mode 7 Asynchronous 16-Bit Burst Write Operations on a 16-Bit Width Device (WRWST=0, WELEN=0 FCLKDIV=1 and ADVHOLD=0, BTMODE = 0)

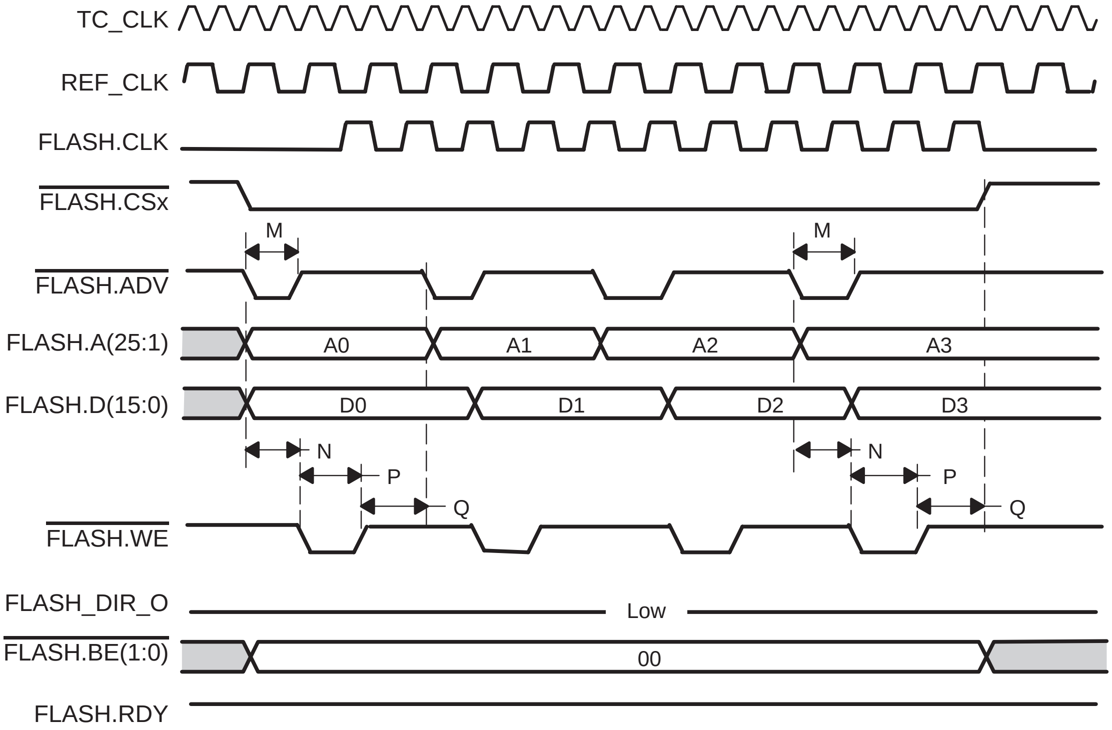
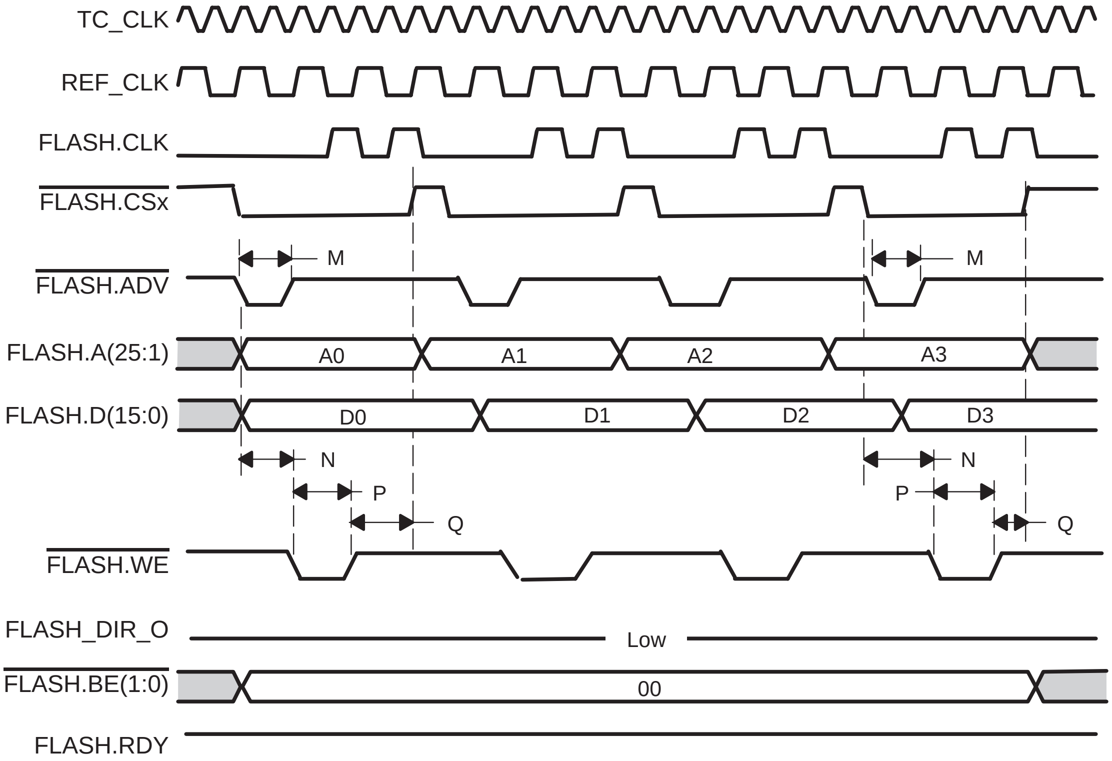


Figure 33. Mode 7 Asynchronous 16-Bit Burst Write Operations on a 16-Bit Width Device (WRWST=0, WELEN=0 FCLKDIV=1 and ADVHOLD=0, BTMODE = 1 and BTWST = 0)



3.2.14 Bus Turn-Around and CS Negation Time Control

- ❑ When slow devices are attached to the IC, it can be necessary to control the next data bus activation time after a read access to this slow device. This data bus activation time can be either an EMIFS write access or an EMIFS read access to a device attached to a different CS. The minimum idle time before next CS activation is two TC_CK cycles for independent access (no split or burst accesses). It can be extended to more than two TC_CK cycles by setting proper value in the BTWST bit field of the CS configuration register attached to the slow device.
  - ■ CS pulse width high = (BTWST +1) TC_CK.
  - ■ BTWST must be at least two to have more than two-TC_CK-cycle idle time.
- ❑ In the case of successive read accesses due to EMIFS access size adaptation, or in the case of burst read access in mode 0, the CS negation time between the successive read accesses is at least one TC_CK cycle. It can also be extended by the BTWST field.

- ❑ After a read completion, if no other access (RD, WR) is pending, the data bus is driven with the previous read value. The bus turn-around time (OE going high to direction going out) is a minimum of 1 TC_CK cycle and can be extended through BTWST.
- ❑ Table 1 shows the bus turn around cycles inserted for various transitions with EMIFS when BTMODE=0.

Table 1. Idle Time Between Different Bus Access Transitions (BTMODE = 0)

Access(n)	Access(n+1)	Chip-Select	Idle Time	Length(BTWST)
RD(csx)	RD(csx)	Same	Inserted	CSX
RD(csx)	WR(csx)	Same	Inserted	CSX
WR(csx)	RD(csx)	Same	Not inserted	−
WR(csx)	WR(csx)	Same	Not inserted	−
RD(csx)	RD(csy) x != y	Different	Inserted	CSX
RD(csx)	WR(csy) x != y	Different	Inserted	CSX
WR(csx)	RD(csy)x != y	Different	Not inserted	−
WR(csx)	WR(csy)x !=y	Different	Not inserted	−

Figure 34. Wait States During a Read to Read Operation (BTWST (CSX) = 2 and BTWST (CSY) = 1, BTMODE = 0)

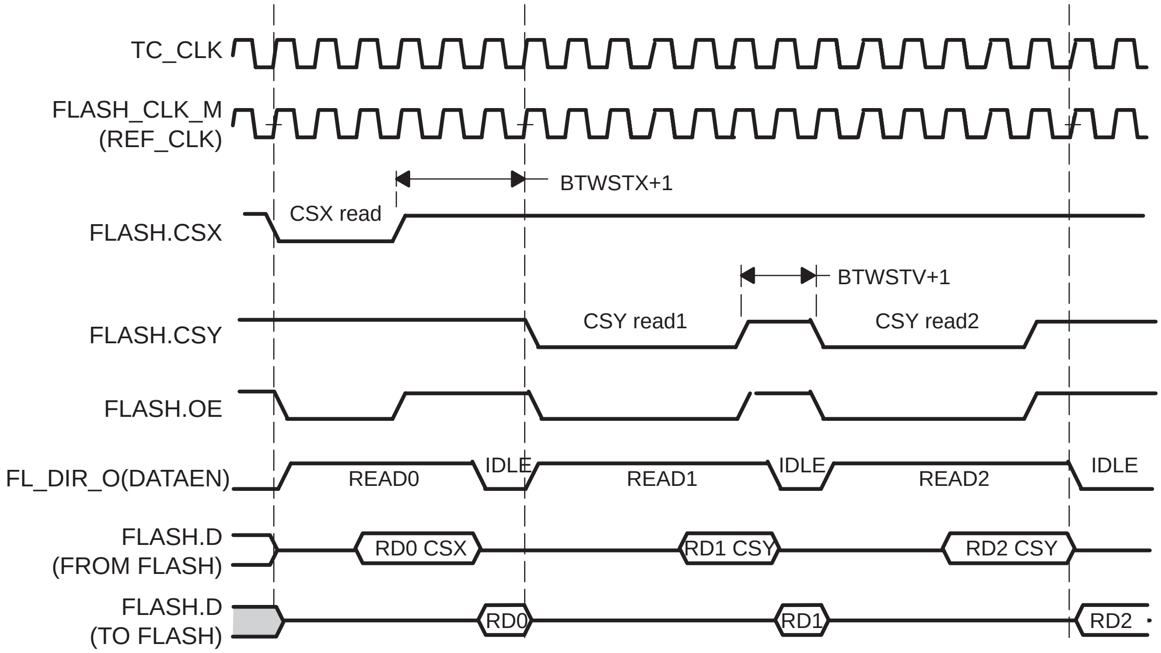
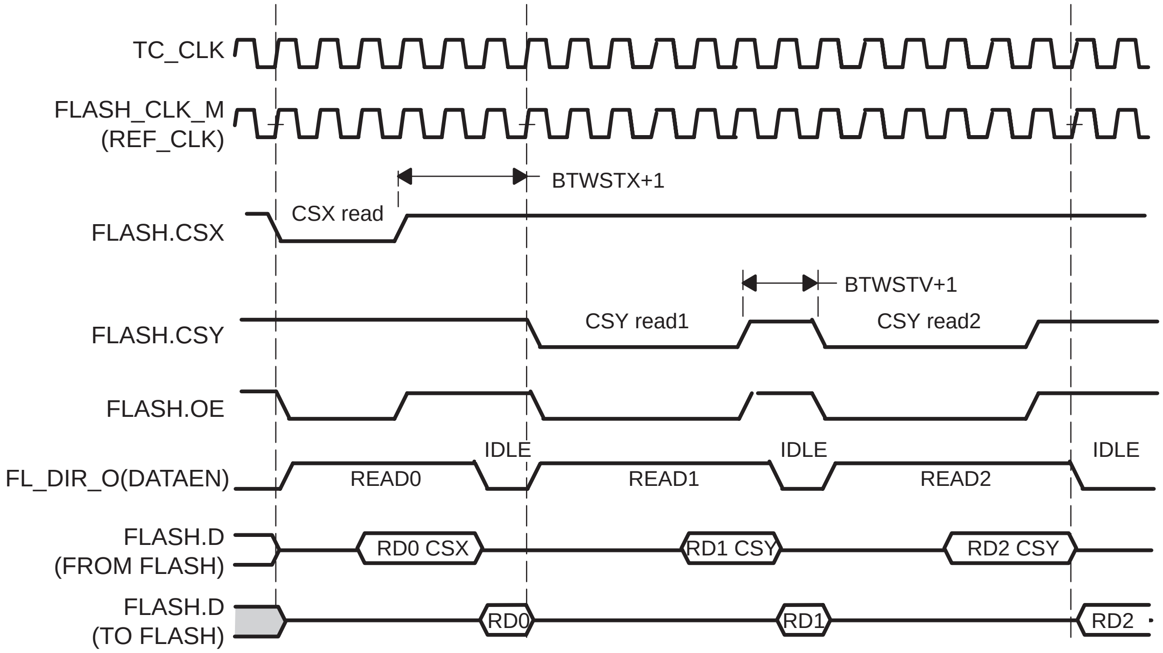


Figure 35. Wait States During a Read to Write Transition (BTWST(CSX) = 3 BTWST (CSY) = 2, BTMODE = 0)



- ❑ The BTMODE field in the advance CS configuration register (see Table 29) extends the previous mode. When BTMODE=1, the BTWST bit field controls CS negation time between any type of successive accesses to the same CS.

- ❑ In the case of successive write accesses due to EMIFS access size adaptation or in the case of write burst read access, there is no CS negation time between successive write accesses unless BTMODE is set. When BTMODE is set, the CS negation time between the successive write accesses is at least one TC_CK cycle and it can be extended by the BTWST field in the CS configuration (see Table 19) register (BTMODE set or clear).

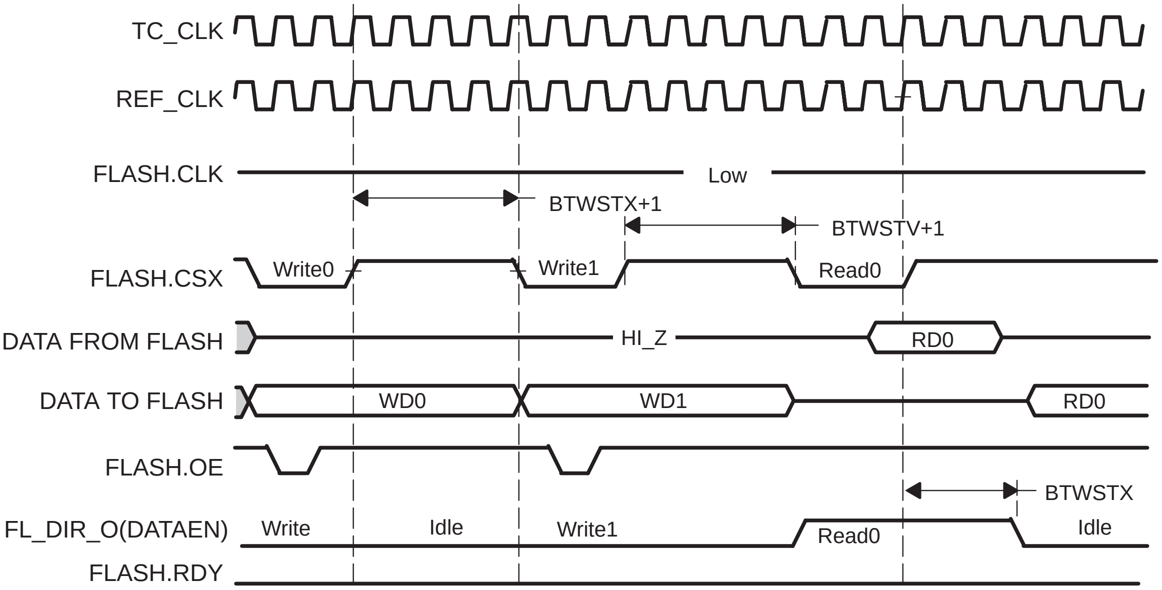
  - ■ CS pulse width high= (BTWST +1) TC_CK

- ❑ Table 2 shows the idle cycles inserted for various transitions with EMIFS when BTMODE=1.

Table 2. Idle Time Between Different Bus Access Transitions (BTMODE = 1)

Access(n)	Access(n+1)	Chip-Select	Idle Time	Length(BTWST)
RD(csx)	RD(csx)	Same	Inserted	CSX
RD(csx)	WR(csx)	Same	Inserted	CSX
WR(csx)	RD(csx)	Same	Inserted	CSX
WR(csx)	WR(csx)	Same	Inserted	CSX
RD(csx)	RD(csy) x != y	Different	Inserted	CSX
RD(csx)	WR(csy) x != y	Different	Inserted	CSX
WR(csx)	RD(csy)x != y	Different	Not inserted	−
WR(csx)	WR(csy)x !=y	Different	Not inserted	−

Figure 36. Wait States During a Write-to-Write and Write-to-Read Transition to Same Chip-Select (BTWST CSX = 3 BTMODE = 1)



3.2.15 External Device Reset Control

- EMIFS interface includes the $\overline{\text{FLASH.RP}}$ output signal. The $\overline{\text{FLASH.RP}}$ output pin is activated during OMAP warm and cold reset. $\overline{\text{FLASH.RP}}$ is also activated when the TC enters the idle state if the RESPWR_EN bit field of the clock and reset ARM_EWUPCT configuration register is set (see Table 73).

- The EXTPWR bit field in the ARM_EWUPCT register is used to specify the minimum time between $\overline{\text{FLASH.RP}}$ deasserted and TC going out of idle state (see section 4.3.4).

3.2.16 Dynamic Auto Idle and System Idle Synchronization

- EMIFS supports auto idle mode (clock gating) to dynamically reduce power consumption when no requests are pending and no accesses are on-going. The dynamic auto idle mode is enabled by setting the PWD_EN bit field of the EMIFS configuration register to 1.

- Upon clock and reset idle request, the EMIFS can send the idle request acknowledge when all on going transactions are completed. This allows the clock and reset module to cut the EMIFS source clock properly. The idle request acknowledge process is enabled by setting the PDE bit field of the EMIFS configuration register to 1 (see Table 18).

3.2.17 Abort Management

The EMIFS can issue an interrupt in two abort scenarios:

- Restricted access mode violation on CS0
  - Access to CS0 address space is limited to the MPU core.
  - When an access to CS0 space is initiated by the DSP/System DMA, the DSP core, or the OCP-I, the EMIFS completed the access giving back a 0 value as read value or without writing the given value to the final destination.
  - The EMIFS raises a TC abort interrupt (IRQ_ABORT), sets the abort flag bit, sets the restricted access error status bit, and updates the host ID bit field in the abort type register. The EMIFS updates the abort address register with the requested address, causing the access violation.
  - The abort flag bit is cleared when the abort type register is read.

- Time-out issue during any access in case of full-handshaking mode
  - This feature can be enabled with the TIMEOUT_EN bit field and with an 8-bit TIMEOUT programmable value (expressed in REF_CLK cycles) in the EMIFS abort time-out register (see Table 12).
  - On beginning of an access, the time-out counter starts counting down. If the counter reaches 0 before the device response, the EMIFS raises an interrupt, sets the abort flag bit, sets the time out error status bit, and updates the host ID bit field in the abort type register. The EMIFS updates the abort address register with the requested address, causing the access time out.
  - The abort flag bit is cleared when the abort type register is read.
  - At reset, TIMEOUT_EN is set to 1 and time-out value is set to 255 REF_CLK cycles.

3.2.18 EMIFS Boot Mode

There are a number of external mechanisms that affect the initial state of the EMIFS during reset. The reset value of three bit fields in the EMIFS configuration (Table 18) and chip-select configuration registers (CS0 and CS3 only, Table 19) changes depending on the state of these external mechanisms. The three bits are BM (boot mode), BW (boot execution memory width, either 16 or 32 bits), and MAD (multiplexed address and data protocol). The reset value of these pins depends on the following:

BM reset value:

- BM is 0 when:
  - MPU_BOOT is 0.

    OR
  - The device type is not emulation.
- BM is 1 when:
  - MPU_BOOT (ball J20) is 1.

    AND
  - The device type is emulation.

BW reset value:

- BW is 0 (boot execution is from a 16−bit wide memory, external boot):
  - BM is 1.
- BW is 1 (boot execution is from a 32−bit wide memory, internal boot):
  - Device type is production type.

    OR
  - The device type is emulation. AND MPU_BOOT is 0.

MAD reset value:

- MAD is 0 (data and address non−multiplexed protocol):
  - Device type is production.

    OR
  - Device type is emulator AND either MPU_BOOT OR GPIO_1 is 0.
- MAD is 1 (data and address multiplexed protocol):
  - Device type is emulation AND MPU_BOOT is 1 AND GPIO_1 is 1.

Note that the internal boot ROM may change the values of the MAD bit depending on the execution path.

When BW = 0, the following CS0 and CS3 configuration is selected.

- RDMODE=0
- FCLKDIV=3
- RDWST=15
- WELEN =15
- WRWST=15
- RT=0

When BW = 1, the following CS0 and CS3 configuration is selected.

- RDMODE=7
- FCLKDIV=0
- PGWST=0
- WRWST=0
- RDWST=0
- RT=1

The CS1 and CS2 reset configuration registers are independent of the boot input pins state.

Table 3. CS1 and CS2 Configuration Register Reset Value

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	PGWST EN	PGWST				BTWST				MAD		BW		RDMODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	PGWST/WELEN				WRWST				RDWST					RT	FCLKDIV	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1

When BM is 0, CS0 is activated in the 0000:0000−03FF:FFFF range and CS3 is activated in the 0C00:0000−0FFF:FFFF range. When BM is 1, CS3 is activated in the 0000:0000−03FF:FFFF range and CS0 is activated in the 0C00:0000−0FFF:FFFF range.

3.3 EMIFF Programming

3.3.1 Main Features

The OMAP EMIFF is an SDRAM controller that manages all accesses by the various initiators of an OMAP-based system.

It can support one 16-bit device or two 8-bit devices. The external interface data bus width is always 16 bits.

The following devices are supported:

- ❑ Standard single-data-rate SDRAM
- ❑ Low-power single-data-rate SDRAM
- ❑ Mobile double-data-rate SDRAM

In terms of capacity and organization for the memory components that can be attached, the EMIFF can handle:

- ❑ 1G-bit, 512M-bit, 256M-bit, 128M-bit, 64M-bit, and 16M-bit devices.
- ❑ 2-bank 16M-bit devices, 2-bank or 4-bank 64M-bit devices, and 4-bank only for any other capacity.
- ❑ x8 (two devices) or x16 (single device) data bus configuration, except for the 1G-bit device. The EMIFF only supports x16 1G-bit device (a single device). The maximum external SDRAM configuration is 128M bytes.

The SDRAM_type field of the EMIFF interface SDRAM configuration register must be used to specify the physical configuration of the devices.

The SDRAM type selection is the first action required from the software driver, using the SDRAM_Type field of the EMIFF SDRAM operation register. Types supported are regular SDR SDRAM, low-power SDRAM, and mobile DDR SDRAM. The SDRAM controller supports:

- ❑ The self-refresh mode (idle), auto refresh, and other operating modes (HPHB, LPLB, and POM0 modes).
- ❑ MRS command and extended MRS command for DDR SDRAM and low-power SDRAM, sent via the SDRAM request manager.
- ❑ For SDR SDRAM, all burst sizes, between 1 and 32 consecutive accesses.
- ❑ For DDR SDRAM, only bursts of 8.
- ❑ Two pipelined levels of request from the SDRAM request manager to enable page interleave timing and reduce overhead cycles by the burst interruption mechanism.

3.3.2 Initialization Sequence

Before it can be accessed for writing or reading after power on, or when exiting deep power-down mode, an SDRAM device must be initialized with various protocol parameters (burst size, CAS idle time, write burst, etc.). This is known in SDRAM literature as the initialization sequence.

Because SDR SDRAM and DDR SDRAM require different initialization sequences, the sequence is manually software driven, using the EMIFF SDRAM manual command register (writing a command opcode into that register generates that command on the SDRAM interface).

The SDRAM operations register should only be programmed after the “a precharge all command” has been issued.

For SDR SDRAM devices, the initialization of the SDRAM bank is accomplished via a preprogrammed series of writes to the command register and the MRS register. There are two possibilities to generate the initialization sequence:

- Writing to the EMIFF SDRAM MRS register (legacy) in the SDRAM controller sets up the device to interface with OMAP. The following SDRAM command sequence is automatically generated:

  1) Precharge all
  2) Load mode register
  3) Autorefresh

  The EMIFF SDRAM MRS register (legacy) is provided for compatibility with existing code and must not be used by new software drivers.

- The new recommended procedure is:

  1) Wait for 200 µs after power up and after the memory clock is running and stable. The software is responsible for this initial wait. During this wait, NOP commands are automatically sent to the memory by default.

  2) Apply a PRECHARGE ALL command by writing to the command register.

  3) Apply two AUTOREFRESH commands by two consecutive writes to the command register.

  4) Apply an MRS LOAD command by writing to the EMIFF SDRAM MRS register (new). It uses the new address, not the legacy address.

  This procedure is also suitable for the mobile DDR device, as this kind of device does not include delay-locked loop technology (DLL).

3.3.3 Memory Mode Registers

The following restrictions apply to the MRS register programming:

- ❑ Burst length must be programmed to full page for SDR devices.
- ❑ For mobile DDR devices, burst length must be programmed to 8 and the CAS idle time must be set to 3.

All SDRAM internal mode registers are mirrored in the EMIFF controller. Standard MRS is provided. EMRS0 is provided for the DDR EMRS register. EMRS1 is provided for low-power SDRAM new features, such as partial array self-refresh, or temperature compensated self-refresh. EMRS2 is provided for future use.

For each register write access, a 12-bit data value is loaded in the memory device to support future new option bits in the existing registers. Always write 0 in all reserved bits.

When reading to these registers, the data in the mirrored register is returned.

3.3.4 EMIFF SDRAM Configuration

The EMIFF SDRAM configuration register must then be programmed to define the other parameters of the interface:

- ❑ Power-down strategy. When the PWD bit is set to 1, the power-down state is automatically entered between memory accesses (see Table 33). When there is no active transaction on the interface, CKE goes low. Any new access request awakes the device before making the access.
- ❑ SDRAM autorefresh control.

To optimize SDRAM bandwidth usage for data transfer, it is preferable to generate a sequence of autorefresh requests, rather than a single autorefresh request. A sequence of four or eight successive autorefreshes can be programmed.

The autorefresh burst request is generated when a 16-bit refresh timer (see SDRAM configuration register, see Table 33) reaches the following user defined values, according to selected SDRAM frequency (via SDRAM configuration register):

- ❑ Memory size and configuration.
- ❑ Entering self-refresh (SLRF bit). The self-refresh mode is automatically exited when the EMIFF receives an access request from OMAP initiators.

The refresh counter value can be set corresponding to system frequency. The following formula can be used for refresh counter value (the counter value is the number of TC_CK between refreshes).

Counter value = (64 ms (refresh rate) / Number of row / t_f) − 50 cycles

50 cycles margin due to a possible transfer currently starting when the refresh request occurs.

Where t_f = 1/system frequency [ns] (system freq = TC_CK freq)

Example: 100-MHz system clock, 4096 rows and 8-burst refresh. t_f = 10 ns.

Counter value = ((64.10^6 ns/4096)/10) − 50

Counter value = 1512 cycles (0x5E8)

If burst autorefresh is selected (4 or 8 consecutive autorefresh commands), this refresh period value is automatically scaled by the hardware accordingly.

3.3.5 EMIFF Configuration Power−Down Considerations

Self−refresh commands will not be issued and the SDRAM clock enable (CKE) will not deactivate under the following conditions:

- EMIFF_CONFIG.pwd = 1
- EMIFF_CONFIG.clk = 0
- EMIFF_CONFIG2.sd_auto_clk = 1

In addition, when in DPD (deep power down) mode, the SLRF bit must not be set, because the CKE cannot operate after DPD exit.

3.3.6 Command Table

Table 4 lists the commands.

Table 4. Command List

Command	$\overline{CS}$	$\overline{RAS}$	$\overline{CAS}$	$\overline{WE}$	CKE	ADDR
Command inhibit (NOP)	H	X	X	X	H	X
No operation (NOP)	L	X	X	X	H	X
Active (select bank and row activate)	L	L	H	H	H	Bank/Row
Read (select bank and column and start read burst)	L	H	L	H	H	Bank/Col
Write (select bank and column and start write burst)	L	H	L	L	H	Bank/Col

Table 4. Command List (Continued)

Command	**$\overline{\text{CS}}$**	**$\overline{\text{RAS}}$**	**$\overline{\text{CAS}}$**	**$\overline{\text{WE}}$**	**CKE**	**ADDR**
Burst stop (for SDR only)	L	H	H	L	H	X
Precharge (deactivate row in a bank or all banks) Autoprecharge is not supported	L	L	H	L	H	A10 Low – Bank selected by BA1 and BA0 (for 4 bank) and BA0 (for 2 bank) High – All banks to be precharged
Autorefresh	L	L	L	H	H	X
Self-refresh Entry	L	L	L	H	H → L	X
Self-refresh Exit	X	X	X	X	L → H	X
MRS, EMRS0, EMRS1 (EMRS1 for M-SDR and M-DDR only)	L	L	L	L	H	Op Code [BA1:BA0] = 01: EMRS0 [BA1:BA0] = 10: EMRS1
Power down entry	H	X	X	X	H → L	X
Power down exit	H	X	X	X	L → H	X
Deep power down entry	L	H	H	L	H → L	X
Deep power down exit	H	X	X	X	L → H	X
Deep power down	The difference between power down and deep power down mode is as follows. In power down mode, whenever there is an access request to EMIFF, the memory is brought out of power down by pulling CKE high. In deep power down mode, even when there is an access request, the memory is not brought out of deep power down mode (CKE is still low). The I/O buffers of the memory are deactivated, and although further commands (after having entered deep power down mode) are accepted by the memory, it does not send or receive any data. After exit from deep power down mode, NOP must be maintained for 200 µs. This must be followed by PRECHARGE ALLPrecharge all command, followed by eight autorefreshes, followed by MRS and EMRS commands.					

X = Don't Care

3.3.7 SDRAM Interface ac Parameters

The 2-bit SDRAM frequency (SDF) field in the EMIFF configuration register allows for selection of a set of ac timings parameters for the interface, depending on the type and speed grade of the memory component used. These parameters are commonly used in the SDRAM literature. As the naming can change from one memory manufacturer to another, the next table provides the functional description of each individual parameter, as named in this specification document.

Table 5. SDRAM AC Parameter Descriptions

Symbol	Description
t_{RAS}	Active to precharge
t_{RC}	Active bank a to active bank a
t_{RFC}	Auto refresh period
t_{RCD}	Active to RD/WR command
t_{RP}	Precharge command period
t_{RRD}	Active bank a to active bank b
t_{WR}	Write recovery time
t_{CCD}	RD/WR command to RD/WR command
t_{CDL}	Last data in to new RD/WR command
t_{MRD}	Load mode register to active/refresh command
t_{XSNR}	Exit self refresh to non-RD command

Four ac configurations are provided for standard SDR memory components, and another set of four configurations is provided for the mobile DDR memory. The selection between these two sets is done automatically, based on the memory type programmed in the EMIFF operation register.

All parameters are individually specified in number of clock cycles, whereas in the memory data sheets, absolute values are used for timing parameters. For a given working frequency, any ac parameter of the selected configuration, once converted into an absolute timing value, must be greater than the equivalent parameter specified by the memory manufacturer.

3.3.8 DLL Control for DDR SDRAM Support

The EMIF fast includes digitally controlled delay technology for interfacing high-speed double-data-rate memory components to meet the strict timing requirements.

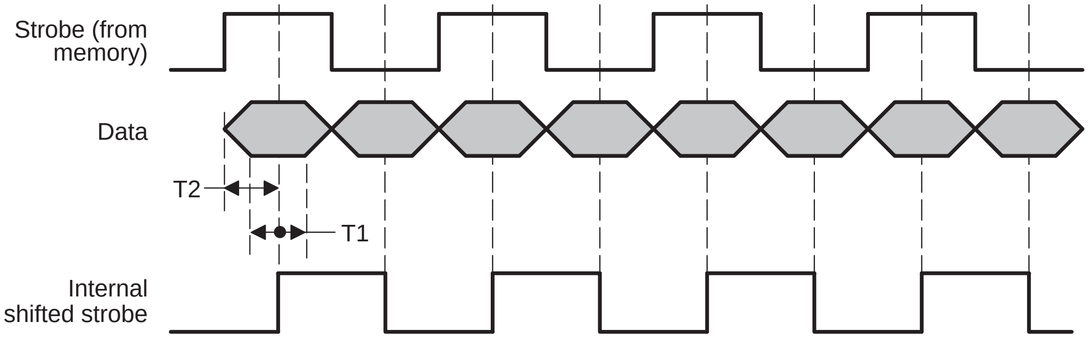
The DLL (delay locked loop) is a calibration module used to track voltage and temperature variations dynamically, as well as to compensate the silicon process dispersion (process voltage temperature (PVT) tracking). Two separate DLL elements are used in the EMIFF: one to control the read timing, and one to control the write timing.

The DLLs control several DCDL companion modules (digitally controlled delay line) by providing an 8-bit value, continuously updated so that it encodes a specific delay value (nominal 72 or 90 degrees) with respect to the memory interface clock frequency in all PVT conditions. The amount of delay added by a DCDL element, controlled by this value under the same PVT conditions, matches the DLL 72 or 90-degree nominal delay.

EMIFF uses two DLLs: one for read operations, the other for write operations.

The read DLL controls two DCDLs. Each DCDL shifts the upper byte strobe or lower byte strobe, respectively, coming from the DDR to ensure the data from the DDR can be properly sampled (see Figure 37).

Figure 37. EMIFF DDR Data Reads (with respect to DLL)

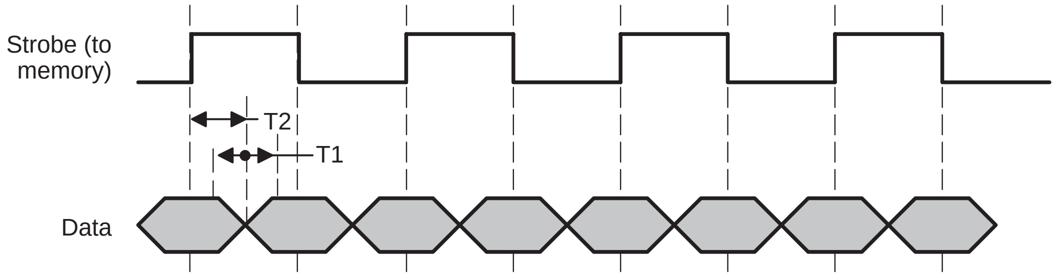


Notes: 1) T1: Fine tune delay control for shift of internal strobe (a manual adjustment). 64 steps of adjustment with a step value of 26.3ps (6 bit signed value in EMIFF_DLL_WRD_CTRL.WO). Note that 26.3 ps is a value valid in a nominal process at room temp and is subject to variation across process and temperature.

2) T2: The EMIFF_DLL_WRD_CTRL .DLLP control bit selects whether the internal strobe is 72 or 90 degrees (20% or 25% delay) from the external strobe.

The write DLL controls one DCDL. This DCDL is used to shift the data lines from OMAP5912 to the DDR (see Figure 38).

Figure 38. EMIFF DDR Data Writes (with respect to DLL)

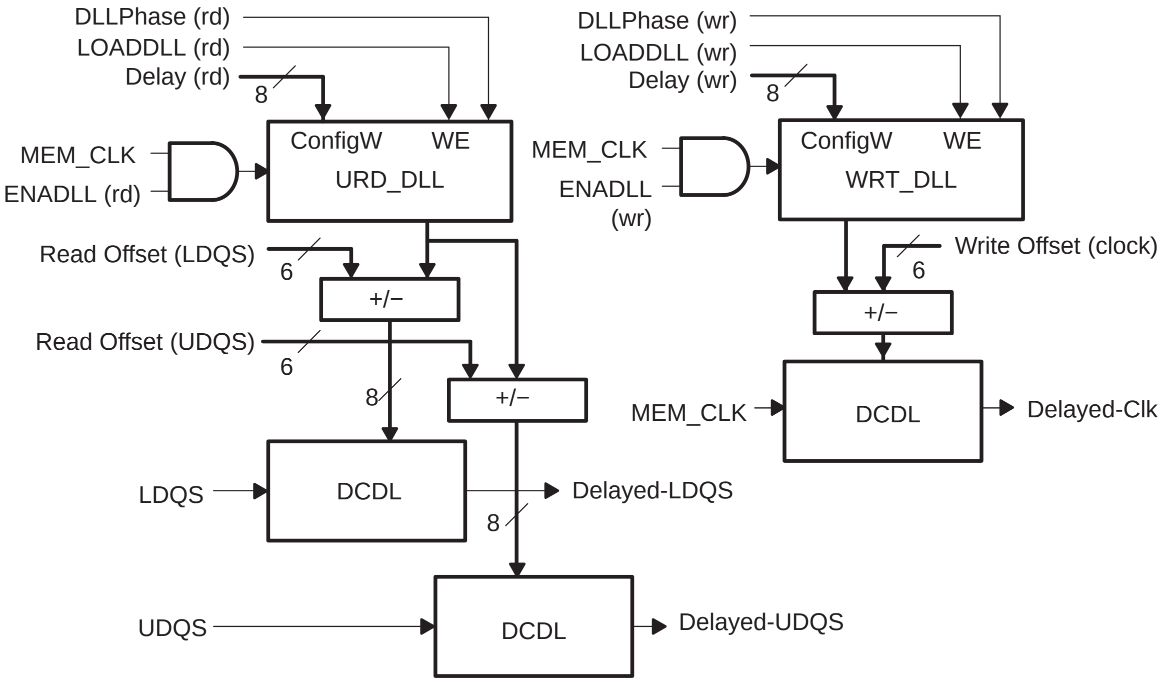


Notes:

1) T1: Fine tune delay control for shifting the data (a manual adjustment). 64 steps of adjustment with a step value of 26.3ps (6 bit signed value in EMIFF_DLL_WRD_CTRL.WO). Note that 26.3 ps is a value valid in a nominal process at room temp and is subject to variation across process and temperature.
2) T2: The DLLPhase control bit selects between whether or not the data is shifted 72 or 90 degrees (20% or 25% delay) from the strobe.

Figure 39 shows the controlled−delay block architecture. This block is configured by the EMIFF DLL control register (bit fields from this register are highlighted using bold characters).

Figure 39. Controlled Delay Subsystem Block Diagram



Thus, there are four important register controls:

- ENADLL in the EMIFF_DLL_WRD_CTRL register (see Table 41) enables or disables DLL.
- DLLPHASE in the EMIFF_DLL_WRD_CTRL register (see Table 41) allows a fine tuning of the delayed transaction to either 72 or 90 degrees out of phase with the internal strobe.
- Delay in the EMIFF_DLL_WRD_CTRL register (see Table 41) is the initial delay value of the strobe. The DLL then tries to lock in on the correct delay. A good initial value is 0x80.
- Write offset in the EMIFF_DLL_WRD_CTRL register (see Table 41) is a manually generated adjustment to the DLL delay. Generally, this field should not be used, as the DLL will lock automatically with the correct delay.

The DLLPHASE control bit is used to set up the nominal delay tracked by the DLL. This delay can be either 72 degrees (20% of the clock period) or 90 degrees (25% of the clock period). The DLL locking range is 75/133 MHz if configured for 90 degrees, and 66/133 MHz if configured for 72 degrees.

The maximum working frequency for the DDR interface is device-dependant. It is limited by the speed of the traffic controller, as well as by the I/O buffer/receiver pair used for the interface (I/O voltage, LVCMOS technology versus SSTL2). The discussion that follows is based on a C035 device, with the traffic controller running at the maximum speed of 96 MHz.

For mobile DDR memories, there is no low-frequency limitation, as the DLL has been removed. The EMIF fast operational range for this type of memory is 0/100 MHz.

The DLL can be loaded with a programmable value. The load capability can be used either to shorten the locking time, by setting an initial value near the expected locked state value, or to assert a given value permanently. In the latter case, the system is said to be working in unlock mode. The unlock mode is useful when interfacing mobile DDR at frequencies below the DLL minimum frequency, i.e. 66 MHz if 72 degrees, or 75 MHz if 90 degrees. Obviously, the accuracy of the delay when locked is much better, but the performance of the unlock mode is good enough below 66 MHz.

The unlock mode is simply set up by asserting the LOADDLL bit of the DLL control register continuously, with the delay field set to the expected value.

When loading a value into the DLL, the DLL must be enabled (ENADLL set to 1) for at least two clock cycles for the load to be effective. To set up the controlled delay block for unlock mode:

- ❑ Write into the DLL control register, with ENADLL set, LOADDLL set, and the delay field set to the expected value.
- ❑ Write into the DLL control register, with ENADLL reset, to save power.

To set up the controlled delay block for lock mode (normal mode):

- ❑ Write into the DLL control register, with ENADLL set, LOADDLL set, and the delay field set to 128 (half range to minimize locking time).
- ❑ Write into the DLL control register, with LOADDLL reset, to launch the tracking process.
- ❑ Wait for at least 336 TC clock cycles to be sure the DLL is locked before accessing the memory.

Once locked, the DLL can be periodically disabled under software control, assuming this period is short compared with the voltage and temperature variations. When disabled, the DLL counter is frozen to its current value. When enabled again, the tracking resumes from that DLL counter value.

The delay can be fine-tuned using the programmable offsets. These offsets are 6-bit signed quantities, resulting in a +31/−32 adjustment range.

Step value: 26.3 ps +/− 10 ns. This value is valid in all PVT conditions (industrial temperature range −40 to 125 degrees, voltage range 1.35V to 1.65V, weak or strong process).

Table 6 provides the typical programming required for an OMAP3.2-based device, C035 process, with a 1.8V−3V I/O voltage range.

Table 6. EMIF Fast Controlled-Delay Block Programming

Memory Type	**DDR 1**	**Mobile DDR**	
Frequency range	83−100 MHz	0−66 MHz	66−100 MHz
DLLPhase (RD DLL/WR DLL)	90/72°	x	90/72°
Unlock mode	No	Yes	No
Forced delay value	x	104	X
Read offset	8	0	8
Write offset	−1	−18	−1

The DLL behavior can be monitored by the software using the EMIFF DLL status registers (URD status and WRD status registers). For each DLL, the current counter value is available (DLLCount field), as well as two status bits: overflow and underflow. Overflow is set when the DLL counter value is 255 during at least one TC clock cycle, and underflow is set when the DLL counter is 0 during at least one TC clock cycle.

These two status bits are both reset if the DLL is disabled (ENADLL is set to 0 in the DLL control register).

3.3.9 Page-Closing Strategy

The SDRAM operation register that selects between SDR and DDR also allows selection of a page-closing strategy. Three variants are available:

❑ High-power/high-bandwidth mode (HPHB)

In this mode of operation, the active page is always left open at the end of access. The EMIFF controller keeps track of the open pages. It then determines if the current access is to an opened page or to a closed page. If it is to an opened page, access proceeds without latency. If it is to a closed page, the currently active page within that internal bank is closed, the necessary page is activated, and the access proceeds after a necessary latency. This mode of operation is useful in situations where the power consumption is not critical, but the bandwidth achieved is important.

❑ Low-power/low-bandwidth mode (LPLB)

In this mode of operation, the page being accessed is always closed at the end of the current access (the controller adds a precharge command after any read or write to close the row). For every access, the page being accessed must be activated at the beginning of the access, which causes extra cycles of latency and lowers the memory bandwidth. Though this may not be an efficient or optimal operating mode, this scheme is very useful in situations where minimizing power consumption is of primary concern. The power consumption is minimized because only the currently accessed page is open and all pages are closed during idle cycles.

- Programmable operating mode (POM0)

  In this mode of operation, a time-out register is associated with each open page (see Table 46, SDRAM operation register). Because the maximum number of simultaneous open pages is four (one per bank), four time-out registers are provided. Upon an access to an open page, the time-out register is set to the user-programmed value and starts to count down. If the counter reaches zero, the controller then closes the inactive page. This mode of operation has better bandwidth performance than the low-power/ low-bandwidth operating mode and does not consume power as heavily as the high-bandwidth/high-power mode.

  The effective time-out value (in SDRAM clock cycles) with respect to the 7-bit programmed value is programmed value x 4 + 3. The minimum value that can be programmed is 4; consequently, the minimum number of cycles before a row is automatically closed is 19.

As mentioned previously, the command register is used during the initialization sequence, but also controls the deep power mode (enter/exit commands) if supported by the memory device.

Arbitration in EMIFF is performed according to the priority algorithms described in section 3.6.

3.4 OCP-I Programming

The OMAP 3.2 OCP external initiator port is an interface intended to connect an external master device to the OMAP 3.2 platform. The OMAP 3.2 core appears as a slave, and its entire memory map, including TIPB peripherals, is accessible. The interconnect bus is compliant with the OCP specification.

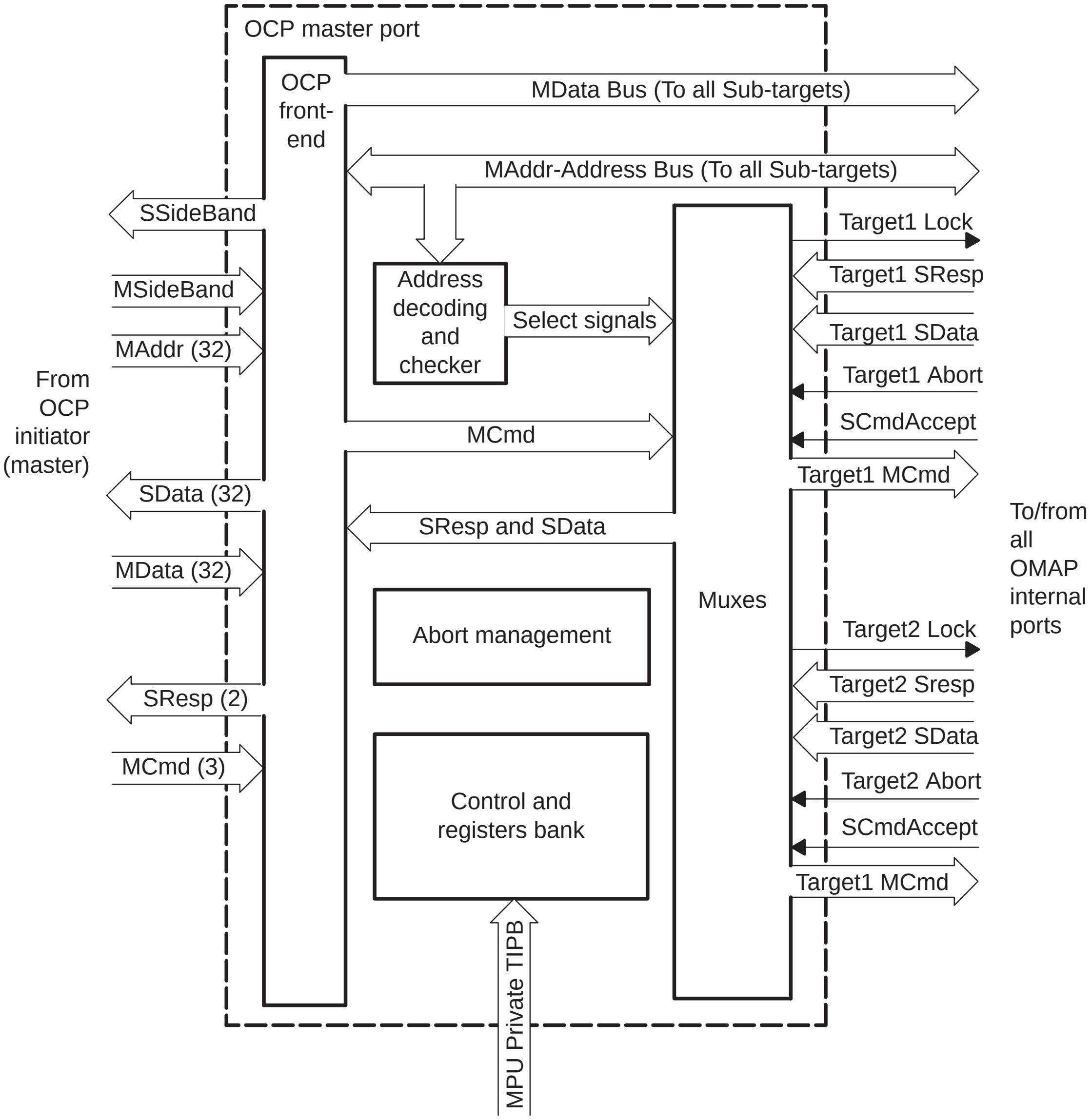
The OCP initiator is capable of single transfers in 8, 16, and 32 bits and burst mode transfers in 32 bits.

OCP-I provides access to the following OMAP targets:

- EMIFS (external slow memories)
- EMIFF (external fast memories)
- OCPT1
- OCPT2
- Multibank OCPT1/2
- MPUI (not part of the TC; see section 5)
- MPU core private TIPB bridge (not part of the TC; see section 7)
- MPU core public TIPB bridge (not part of the TC; see section 7)

Figure 40 shows the OCP-I block diagram.

Figure 40. OCP-I Block Diagram



Though OCP-I has visibility to all OMAP resources, a protection register allows validation of the access permission on a per target basis. Any access to an unauthorized resource generates an abort.

The MPU core controls the OCP-I port through its private TIPB; if permission is given to the external initiator to access the private TIPB, the external master also accesses the OCP-I configuration registers.

If OS protection is needed, it is recommended that any device integrating the OMAP 3.2 core provide an MMU or a translation table controlled by the OMAP MPU core. Then it is the responsibility of the system software to ensure that undefined or unauthorized locations are not accessed.

An internal address checker continuously compares the OCP address bus and the OMAP 3.2 memory map. An abort signal is provided if the transaction is not correct.

OCP-I contains other status and control registers listed in Table 7.

3.4.1 Address and Command Fault Registers

When a non-OMAP-defined address is detected or a bus error occurs from the subtargets on a write access, an abort is sent to the initiator to indicate that the current access cannot be completed. The access to the OMAP subtarget is terminated and the address and command buses are stored in the fault registers.

3.4.2 Abort Type Register

The type of the abort event is reported in a status register.

The interrupt handler can clear this register and the interrupt request by reading any value to this register.

3.4.3 Protection Register

This register provides access protection to the following targets:

- EMIF-slow and EMIF-fast
- OCPT1 and OCPT2
- Multibank OCPT1/2
- MPUI
- MPU core private TIPB bridge and MPU core public TIPB bridge

The reset value of the protection register bits is determined by the value of the static_reset_protect_mode pin.

3.5 Traffic Controller Registers

TC registers are distributed into the traffic controller submodules:

- OCP-T1/OCP-T2 registers
- EMIFS registers

- ❑ EMIFF registers
- ❑ OCPI registers
- ❑ TC Abort management registers (not available on XOMAP5912 or POMAP5912 devices)

3.5.1 OCP-T1/OCT-T2 Registers

Table 7 lists the 32-bit OCP-T1 and OCP-T2 registers. Table 7 through Table 15 describe the register bits.

Table 7. OCP-T1/OCP-T2 Registers

Base Address = FFFE CC00			
Name	**Description**	**R/W**	**Offset**
OCPT1_PRIOR	OCP-T1 LRU priority register	R/W	0x00
OCPT1_PTOR1	OCP-T1 dynamic priority time-out 1	R/W	0xA0
OCPT1_PTOR2	OCP-T1 dynamic priority time-out 2	R/W	0xA4
OCPT1_PTOR3	OCP-T1 dynamic priority time-out 3	R/W	0xA8
OCPT1_ATOR	OCP-T1 abort time-out	R/W	0xAC
OCPT1_AADDR	OCP-T1 abort address	R	0xB0
OCPT1_ATYPER	OCP-T1 abort type	R	0xB4
OCPT_CONFIG_REG	OCP target configuration register	R/W	0xB8
OCPT2_PRIOR	OCP-T2 LRU priority register	R/W	0xD0
OCPT2_PTOR1	OCP-T2 dynamic priority time-out 1	R/W	0xD4
OCPT2_PTOR2	OCP-T2 dynamic priority time-out 2	R/W	0xD8
OCPT2_PTOR3	OCP-T2 dynamic priority time-out 3	R/W	0xDC
OCPT2_ATOR	OCP-T2 abort time-out	R/W	0xE0
OCPT2_AADDR	OCP-T2 abort address	R	0xE4
OCPT2_ATYPER	OCP-T2 abort type	R	0xE8

Table 8. OCP Priority Registers 1 and 2(OCPT1_PRIOR and OCPT2_PRIOR)

Base Address = 0xFFFE CC00, Offsets = 0x00 and 0xD0				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:16	RESERVED	Reserved. To ensure software compatibility, a reserved bit should be written to 0 and read value should be considered undefined.	R/W	0x0000
15:12	OCP_PRIORITY	Number of consecutive accesses allowed for OCP-I.	R/W	0000

Table 8. OCP Priority Registers 1 and 2(OCPT1_PRIOR and OCPT2_PRIOR) (Continued)

Base Address = 0xFFFE CC00, Offsets = 0x00 and 0xD0				
Bit	**Name**	**Function**	**R/W**	**Reset**
11:8	DMA_PRIORITY	Number of consecutive accesses allowed for DSP/System DMA.	R/W	0000
7		Reserved.		
6:4	DSP_PRIORITY	Number of consecutive accesses allowed for DSP core.	R/W	000
3		Reserved.		
2:0	ARM_PRIORITY	Number of consecutive accesses allowed for MPU core.	R/W	000

The OCP target priority registers (OCPTx_PRIOR) allow the target to give consecutive accesses to a host when the host is granted the OCP target.

The MPU core and DSP core can have from 1 to 8 consecutive accesses. The DSP/System DMA and the OCP initiator can have 1 to 16 consecutive accesses, depending on the content of corresponding bits in their priority registers.

Table 9. OCP-T1 and OCP-T2 Priority Time-Out Registers 1 (OCPT1_PTOR1 and OCPT2_PTOR1)

Base Address = 0xFFFE CC00, Offsets = 0xA0 and 0xD4				
Bits	**Field**	**Description**	**R/W**	**Reset**
31:8	RESERVED	Reserved. To ensure software compatibility, a reserved bit should be written to 0.	R/W	0x000000
7:0	DMA	Number of TC_CK cycles that DSP/System DMA must wait in low-priority queue before going to high-priority queue	R/W	0x00

Table 10. OCP-T1 and OCP-T2 Priority Time-Out Registers 2 (OCPT1_PTOR2 and OCPT2_PTOR2)

Base Address = 0xFFFE CC00, Offsets = 0xA4 and 0xD8				
Bits	**Field**	**Description**	**R/W**	**Reset**
31:24	RESERVED	Reserved. To ensure software compatibility, a reserved bit should be written to 0.	R/W	0x00
23:16	DSP	Number of TC_CK cycles that DSP core must wait in low-priority queue before going to high-priority queue	R/W	0x00

Table 10. OCP-T1 and OCP-T2 Priority Time-Out Registers 2 (OCPT1_PTOR2 and OCPT2_PTOR2) (Continued)

Bits	Field	Description	R/W	Reset
15:8	RESERVED	Reserved. To ensure software compatibility, a reserved bit should be written to 0 and read value should be considered undefined.	R/W	0x00
7:0	LCD	Number of TC_CK cycles that LCD must wait in low-priority queue before going to high-priority queue	R/W	0x00

Table 11. OCP-T1 and OCP-T2 Priority Time-Out Registers 3 (OCPT1_PTOR3 and OCPT2_PTOR3)

Base Address = 0xFFFE CC00, Offsets = 0xA8 and 0xDC				
Bits	**Field**	**Description**	**R/W**	**Reset**
31:8	RESERVED	Reserved. To ensure software compatibility, a reserved bit should be written to 0 and read value should be considered undefined.	R/W	0x000000
7:0	OCP-I	Number of TC_CK cycles that OCP-I must wait in low-priority queue before going to high-priority queue.	R/W	0x00

The priority time-out registers (OCPTx_PTORy) are used to store the number of TC_CK clock cycles before DSP core, DSP/System DMA, LCD, or OCPI requests are made high priority in the dynamic priority scheme for the OCP target.

Table 12. OCP-T1 and OCP-T2 Abort Time-Out Registers (OCPT1_ATOR and OCPT2_ATOR)

Base Address = 0xFFFE CC00, Offsets = 0xAC and 0xE0				
Bits	**Field**	**Description**	**R/W**	**Reset**
31:9	RESERVED	Reserved. To ensure software compatibility, a reserved bit should be written to 0 and read value should be considered undefined.	R/W	0x000000
8	TIMEOUT_EN	Enable time-out bit. 0: Disable. 1: Enable.	R/W	1
7:0	TIMEOUT	Number of counted-down clock cycles before sending out abort signal if there is no response from the slave.	R/W	0xFF

The abort time-out register (ATOR) is used to store the number of clock cycles the OCP target counts down before activating the abort signal because the peripheral did not return a SRESP signal.

Table 13. OCP-T1 and OCP-T2 Abort Address Registers—Access Address (OCPT1_AADDR and OCPT2_AADDR)

Base Address = 0xFFFE CC00, Offsets = 0xB0 and 0xE4				
Bits	**Field**	**Description**	**R/W**	**Reset**
31:0	Address	Address of access that caused abort.	R	0x0000 0000

Table 14. OCP-T1 and OCP-T2 Abort Type Registers—Access Address (OCPT1_ATYPER and OCPT2_ATYPER)

Base Address = 0xFFFE CC00, Offsets = 0xB4 and 0xE8				
Bits	**Field**	**Description**	**R/W**	**Reset**
31:5	RESERVED	Reserved. To ensure software compatibility, a reserved bit should be written to 0 and read value should be considered undefined.	R/W	0
4	TIMEOUT_ERR	Abort generated by time-out register. 0: No abort. 1: Abort.	R	0
3	BUS_ERR	Abort generated by error coming from external peripherals 0: No abort. 1: Abort.	R	0
2:1	HOST_ID	Host ID of request that caused memory fault 00: MPU core 01: DSP core 10: DMA 11: OCPI	R	00
0	ABORT_FLAG	0: No abort. 1: Abort.	R	0

Table 15. OCP Target Configuration Register (OCPT_CONFIG_REG)

Base Address = 0xFFFE CC00, Offset = 0xB8				
Bits	**Field**	**Description**	**R/W**	**Reset**
31:2	Reserved	Reserved, must be all 0s.	R/W	0
1	PIPELN_RD_EN	0: Pipeline read operation disabled. 1: Pipeline read operation enabled.	R/W	1
0	AUTO_GATED_CLK	0: Autogating clock disabled. 1: Autogating clock feature enabled to save power.	R/W	0

This register contains the PIPELN_RD_EN bit that enables or disables the pipeline read operation in OCP Target. When enabled, the OCP target will start the next read command when the accept for the previous read command arrives.

3.5.2 EMIFS Registers

Table 16 lists the 32-bit EMIFS registers. Table 17 through Table 29 describe the register bits.

Table 16. EMIFS Registers

Base Address = FFFE CC00			
Name	**Description**	**R/W**	**Offset**
EMIFS_PRIOR	EMIFS LRU priority register	R/W	0x04
EMIFS_CONFIG	EMIFS configuration register	R/W	0x0C
EMIFS_CCS0	EMIFS chip-select configuration CS0	R/W	0x10
EMIFS_CCS1	EMIFS chip-select configuration CS1	R/W	0x14
EMIFS_CCS2	EMIFS chip-select configuration CS2	R/W	0x18
EMIFS_CCS3	EMIFS chip-select configuration CS3	R/W	0x1C
EMIFS_PTOR1	EMIFS dynamic priority time-out 1	R/W	0x28
EMIFS_PTOR2	EMIFS dynamic priority time-out 2	R/W	0x2C
EMIFS_PTOR3	EMIFS dynamic priority time-out 3	R/W	0x30
EMIFS_DWS	EMIFS dynamic wait states	R/W	0x40

Note: The EMIFS chip-select configuration register reset values depend on the input boot pin state at IC reset release time. For more details, see section 3.2.18, *EMIFS Boot Mode*.

Table 16. EMIFS Registers (Continued)

Base Address = FFFE CC00			
Name	**Description**	**R/W**	**Offset**
EMIFS_AADDR	EMIFS abort address	R	0x44
EMIFS_ATYPER	EMIFS abort type	R	0x48
EMIFS_ATOR	EMIFS abort time-out	R/W	0x4C
EMIFS_ACS0	Advanced EMIFS chip-select configuration nCS0	R/W	0x50
EMIFS_ACS1	Advanced EMIFS chip-select configuration nCS1	R/W	0x54
EMIFS_ACS2	Advanced EMIFS chip-select configuration nCS2	R/W	0x58
EMIFS_ACS3	Advanced EMIFS chip-select configuration nCS3	R/W	0x5C

Note: The EMIFS chip-select configuration register reset values depend on the input boot pin state at IC reset release time. For more details, see section 3.2.18, *EMIFS Boot Mode*.

Table 17. EMIFS Priority Register (EMIFS_PRIOR)

Base Address = 0xFFFE CC00, Offset = 0x04				
Bits	**Field**	**Description**	**R/W**	**Reset**
31:16	RESERVED	Reserved. To ensure software compatibility, a reserved bit should be written to 0 and read value should be considered undefined.	R/W	0x0000
15:12	OCPI	OCPI consecutive access	R/W	0x0
11:8	DMA	DSP/System DMA consecutive access	R/W	0x0
7	RESERVED	Reserved	R/W	0
6:4	DSP	DSP core consecutive access	R/W	0
3	RESERVED	Reserved	R/W	0
2:0	MPU	MPU core consecutive access	R/W	000

The EMIFS priority register allows the EMIFS to give consecutive accesses to a master when EMIFS is configured for least recently used (LRU) priority arbitration. The MPU core and the DSP core can have 0 to 7 consecutive accesses, whereas the DSP/System DMA and the OCP-I can have 0 to 15 consecutive accesses, according to the content of the corresponding bits. For burst accesses, a four 32-bit burst access is considered as one access.

Table 18. EMIFS Configuration Register (EMIFS_CONFIG)

Base Address = 0xFFFE CC00, Offset = 0x0C				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:5	RESERVED	Reserved. To ensure software compatibility, a reserved bit should be written to 0 and read value should be considered undefined.	R/W	0x0000000
4	FR	Ready signal. This bit is a copy of the Ready input pin sample by TC_CK (2 TC_CK cycles delay from input pin to register update). 0: Ready pin is low. 1: Ready pin is high.	R	ND
3	PDE	System power-down acknowledge 0: Acknowledge disabled. 1: Acknowledge enabled.	R/W	0
2	PWD_EN	Dynamic auto idle 0: Disable. 1: Enable.	R/W	0
1	BM	Boot mode. Enables CS0 and CS3 address decoding swapping (see section 3.2.18 for BM reset value.) 0: CS0 [0000:0000 − 03FF:FFFF] CS3 [0C00:0000 − 0FFF:FFFF] 1: CS0 [0C00:0000 − 0FFF:FFFF] CS3 [0000:0000 − 03FF:FFFF]	R/W	See note
0	WP	Write protect output pin control 0: WP output pin is set low. 1: WP output pin is set high.	R/W	0

Note: Reset value depends on external pins (see section 3.2.18).

Table 19. EMIFS Chip-Select Configuration Registers (EMIFS_CCS0, EMIFS_CCS1,...,EMIFS_CCS3)

Base Address = 0xFFFE CC00, Offsets = 0x10, 0x14, 0x18, 0x1C				
Bit	**Field**	**Description**	**R/W**	**Reset**
31	PGWSTEN	0: PGWST is specified by 15:12. 1: PGWST is specified by 30:27.	R/W	0
30:27	PGWST	When PGWSTEN is 1, this bit controls the wait states cycle number between accesses in a page for asynchronous page mode.	R/W	0000

Note: The EMIFS chip-select configuration register reset values for CS0 and CS3 depend on a number of factors at reset release time. For more details, see section 3.2.18.

Table 19. EMIFS Chip-Select Configuration Registers (EMIFS_CCS0, EMIFS_CCS1,...,EMIFS_CCS3) (Continued)

Base Address = 0xFFFE CC00, Offsets = 0x10, 0x14, 0x18, 0x1C				
Bit	**Field**	**Description**	**R/W**	**Reset**
26:23	BTWST	Controls the IDLE cycle number for bus turn-around and CS high-pulse-width timing.	R/W	0000
22	MAD	Enables EMIFS multiplexed address and data bus protocol (see section 3.2.18 for MAD reset value). 0: Non-multiplexed protocol. 1: Multiplexed protocol.	R/W	See note
21	RESERVED	Must be written to 0.	R/W	0
20	BW	Controls the data bus width used for this CS (see section 3.2.18 for BW reset value for CS0 and CS3). 0: Data bus is 16 bits wide. 1: Data bus is 32 bits wide.	R/W	1, or see note
19	RESERVED	Reserved for RDMODE expansion. Read value should be considered undefined.	R	ND
18:16	RDMODE	Read mode select (see table below and section 3.2.1). See section 3.2.18 for RDMODE reset value for CS0 and CS3.	R/W	See note or 000
15:12	PGWST/WELEN	Controls the wait states cycle number between accesses in a page for asynchronous page mode. Controls the WE pulse length during a write access. When PGWSTEN is 0, this bit specifies both PGWST/WELEN When PGWSTEN is 1, this bit specifies only WELEN	R/W	See note or 1111
11:8	WRWST	Controls the wait states cycle number for write operation.	R/W	See note or 1111
7:4	RDWST	Controls the wait states cycle number for asynchronous read operation and the initial idle time for asynchronous read page mode and synchronous read mode.	R/W	See note or 1111
3	RESERVED	Reserved. Writing to this bit has no effect. Reading it returns undefined value.	R/W	1

Note: The EMIFS chip-select configuration register reset values for CS0 and CS3 depend on a number of factors at reset release time. For more details, see section 3.2.18.

Table 19. EMIFS Chip-Select Configuration Registers (EMIFS_CCS0, EMIFS_CCS1,...,EMIFS_CCS3) (Continued)

Base Address = 0xFFFE CC00, Offsets = 0x10, 0x14, 0x18, 0x1C				
Bit	**Field**	**Description**	**R/W**	**Reset**
2	RT	Enable the read retimed protocol. This bit may be 1 only in RDMODE 4,5 and 7 only. The system will hang if the retiming bit is set in other modes. See section 3.2.18 for RT reset value for CS0 and CS3. 0: Non retimed protocol. 1: retimed protocol.	R/W	See note or 1
1:0	FCLKDIV	Controls the TC_CK divider. REF_CLK. 00: REF_CLK = TC_CK divide by 1. 01: REF_CLK = TC_CK divide by 2. 10: REF_CLK = TC_CK divide by 4. 11: REF_CLK = TC_CK divide by 6. See section 3.2.18 for RT reset value for CS0 and CS3.	R/W	See note or 11

Note: The EMIFS chip-select configuration register reset values for CS0 and CS3 depend on a number of factors at reset release time. For more details, see section 3.2.18.

Table 20. EMIFS Chip-Select Configuration Register RDMODE Field Definition

RDMODE	Memory
000	Mode 0: Asynchronous read
001	Mode 1: Page mode ROM read—4 words per page
010	Mode 2: Page mode ROM read—8 words per page
011	Mode 3: Page mode ROM read—16 words per page
100	Mode 4: Synchronous burst read mode
101	Mode 5: Synchronous burst read mode
110	Reserved for future extension
111	Mode 7: Synchronous burst read mode

Table 21. EMIFS Time-Out Register 1 (EMIFS_PTOR1)

Base Address = 0xFFFE CC00, Offset = 0x28				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:8	RESERVED	Reserved	R/W	0
7:0	DMA	Number of TC_CK cycles	R/W	0

Table 22. EMIFS Time-Out Register 2 (EMIFS_PTOR2)

Base Address = 0xFFFE CC00, Offset = 0x2C				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:8	RESERVED	Reserved	R/W	0
7:0	DSP	Number of TC_CK cycles	R/W	0

Table 23. EMIFS Time-Out Register 3 (EMIFS_PTOR3)

Base Address = 0xFFFE CC00, Offset = 0x30				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:8	RESERVED	Reserved	R/W	0
7:0	OCPI	Number of TC_CK cycles	R/W	0

Time-out registers 1−3 are used to control the number of TC clock cycles before DSP core, DSP/System DMA, or OCP requests are made high priority in the dynamic priority scheme used inside TC.

Some Flash devices deassert the FLASH.RDY (not ready) during write accesses. To accommodate these devices, read accesses must take place in full handshaking mode, while write accesses must take place with full handshaking turned off.

The WRRDYMASK bit field in the EMIFS Dynamic Wait States Control Register (EMIFS_DWS) masks FLASH.RDY during write accesses. This is not available on XOMAP5912 or POMAP5912 devices. When WRRDYMASK = 0, the EMIFS monitors the FLASH.RDY pin during write accesses according to the handshaking mode (OMAP3.2 compatibility mode). When WRDYMASK = 1, the EMIFS ignores the FLASH.RDY pin during write accesses irrespective of the handshaking mode.

Table 24. EMIFS Dynamic Wait States Control Register (EMIFS_DWS)

Base Address = 0xFFFE CC00, Offset = 0x40				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:12	Reserved	Reserved.	R/W	0x000000
11	WRRDYMASK_CS3	Masks FLASH.RDY during a write operation. WRRDYMASK_CS3 is not available on the X and POMAP5912 devices; therefore, bits 8−11 remain reserved on pre-production silicon. 0: The FLAS.RDY signal is monitored during a write access if full-handshaking mode is enabled. 1: Masks the FLASH.RDY signal during a write access irrespective of full-handshaking mode.	R/W	0
10	WRRDYMASK_CS2	Masks FLASH.RDY during a write operation 0: The FLAS.RDY signal is monitored during a write access if full-handshaking mode is enabled. 1: Masks the FLASH.RDY signal during a write access irrespective of full-handshaking mode.	R/W	0
9	WRRDYMASK_CS1	Masks FLASH.RDY during a write operation 0: The FLAS.RDY signal is monitored during a write access if full-handshaking mode is enabled. 1: Masks the FLASH.RDY signal during a write access irrespective of full-handshaking mode.	R/W	0

Table 24. EMIFS Dynamic Wait States Control Register (Continued)(EMIFS_DWS)

Base Address = 0xFFFE CC00, Offset = 0x40				
Bit	**Field**	**Description**	**R/W**	**Reset**
8	WRRDYMASK_CS0	Masks FLASH.RDY during a write operation 0: The FLAS.RDY signal is monitored during a write access if full-handshaking mode is enabled. 1: Masks the FLASH.RDY signal during a write access irrespective of full-handshaking mode.	R/W	0
7	Full handshake enable for CS3	Enables handshaking mode for CS3 0: Full-handshaking. 1: Non-full-handshaking.	R/W	0
6	Full handshake enable for CS2	Enables handshaking mode for CS2 0: Full-handshaking. 1: Non-full-handshaking.	R/W	0
5	Full handshake enable for CS1	Enables handshaking mode for CS1 0: Full-handshaking. 1: Non-full-handshaking.	R/W	0
4	Full handshake enable for CS0	Enables handshaking mode for CS0 0: Full-handshaking. 1: Non-full-handshaking.	R/W	0
3	Dynamic wait states enable for CS3	Enables dynamic wait states mode for CS3 0: Dynamic wait states mode disabled. 1: Dynamic wait states mode enabled.	R/W	0
2	Dynamic wait states enable for CS2	Enables dynamic wait states mode for CS2 0: Dynamic wait states mode disabled. 1: Dynamic wait states mode enabled.	R/W	0
1	Dynamic wait states enable for CS1	Enables dynamic wait states mode for CS1 0: Dynamic wait states mode disabled. 1: Dynamic wait states mode enabled.	R/W	0
0	Dynamic wait states enable for CS0	Enables dynamic wait states mode for CS0 0: Dynamic wait states mode disabled. 1: Dynamic wait states mode enabled.	R/W	0

This register controls if EMIFS has dynamic wait states by using the ready signal. The WRRDYMASK bit allows a full-handshaking read access concurrently with a non-full-handshaking write access. The possible combinations are given in Table 25. The reset value of these bits is 0 to ensure compatibility with OMAP3.2.

Table 25. Read and Write Access Handshaking

Full Handshake Enable bit for CSn	WRRDYMASK Bit for CSn	Handshaking Mode for CSn
0	0	FLASH.RDY is monitored for both read and write accesses (full handshaking)
0	1	FLASH.RDY is monitored for read access (full-handshaking) and masked for write access (non-full-handshaking)
1	x	FLASH.RDY is ignored for both read and write accesses (non-full-handshaking)

Table 26. EMIFS Abort Address Register (EMIFS_AADDR)

Base Address = 0xFFFE CC00, Offset = 0x44				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:0	AA	Abort address	R	0x00000000

This register holds the address involved in the aborted transaction.

Table 27. EMIFS Abort Type Register (EMIFS_ATYPER)

Base Address = 0xFFFE CC00, Offset = 0x48				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:5	Reserved	Reserved. To ensure software compatibility, a reserved bit should be write to 0 and read value should be considered undefined.	R/W	0x0000000
4	TOE	Time out error. Time out status bit set to 1 if abort was caused by a time-out error.	R	0
3	RAE	Restricted access error. Restricted access status bit is set to 1 if abort was caused by a restricted access error.	R	0
2:1	HID	Host ID. Specify the source of the aborted transaction: 00: MPU core 01: DSP core 10: DMA 11: OCP-I	R	00
0	ABORT_FLAG	Abort status bit. Reading the abort type register reset the abort status bit. 0: No abort. 1: Abort.	R	0

This register reports the type of the transaction that has been aborted.

Table 28. EMIFS Abort Time-Out Register (EMIFS_ATOR)

Base Address = 0xFFFE CC00, Offset = 0x4C				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:9	Reserved	Reserved. To ensure software compatibility, a reserved bit should be write to 0 and read value should be considered undefined.	R/W	0x0000000
8	TIMEOUT_EN	Enable the time-out timer. 0: Timer is disabled. 1: Timer is enabled.	R/W	1
7:0	TIMEOUT	Time out counter value in REF_CLK clock cycles.	R/W	0xFF

Table 29. Advanced EMIFS Chip-Select Configuration Registers (EMIFS_ACS0, EMIFS_ACS1,...,EMIFS_ACS3)

Base Address = 0xFFFE CC00, Offset = 0x50, 0x54, 0x58, 0x5C					
Bit	**Field**	**Value**	**Description**	**R/W**	**Reset**
31:12	Reserved		Reserved. To ensure software compatibility, a reserved bit should be write to 0 and read value should be considered undefined.	R/W	0x000000
11	READY_CONFIG		READY_CONFIG can be considered as valid one cycle ahead of or in the same cycle as the data phase with which it is supposed to handshake. READY_CONFIG is not available on the X or POMAP5912 devices and should be treated as reserved for those versions of the silicon.	R/W	0
		0	Ready is monitored one clock cycle ahead of the data phase		
		1	Ready is monitored in the same cycle as the data phase.		
10	CLKMASK		Clock mask is not available on X or POMAP5912 devices.	R/W	0
		0	FLASH.CLK toggles during write operations for RDMODE 4 and 5.		
		1	FLASH.CLK is driven low during write operations for RDMODE 4 and 5.		

Table 29. Advanced EMIFS Chip-Select Configuration Registers (EMIFS_ACS0, EMIFS_ACS1,...,EMIFS_ACS3) (Continued)

Bit	Field	Value	Description	R/W	Reset
9	BTMODE		Enables extended BTWST usage 0: Bus turn around control and RD to RD/WR same CS pulse width high control. 1: Bus turn around control and RD/WR to RD/WR same CS pulse width high control.	R/W	0
8	ADVHOLD		Controls the ADV pulse width low.	R/W	0
7:4	OEHOLD		Controls the number of cycles from OE high to CS high.	R/W	0x0
3:0	OESETUP		Controls the number of cycles inserted from CS low to OE low. When the MAD reset value is 1, the reset value of OESETUP is 0x2 (see section 3.2.18 for MAD reset).	R/W	0x0 or 0x2

3.5.3 EMIFF Registers

Table 30 lists the 32-bit EMIFF registers. Table 32 through Table 57 describe the register bits.

Table 30. EMIFF Registers

Base Address = 0xFFFE CC00			
Name	**Description**	**R/W**	**Offset**
EMIFF_PRIOR	EMIFF priority register	R/W	0x08
EMIFF_CONFIG	EMIFF configuration register	R/W	0x20
EMIFF_MRS†	EMIFF SDRAM MRS register (legacy)	R/W	0x24
EMIFF_CONFIG2	EMIFF configuration register 2	R/W	0x3C
EMIFF_DLL_WRD_CTRL	DLL WRD control register (write byte)	R/W	0x64
EMIFF_DLL_WRD_STAT	DLL WRD status register	R	0x68
EMIFF_MRS_NEW†	EMIFF SDRAM MRS register (new)	R/W	0x70
EMIFF_EMRS0	EMIFF SDRAM EMRS0 register	R/W	0x74
EMIFF_EMRS1	EMIFF SDRAM EMRS1 register	R/W	0x78

† EMIFF_MRS is a legacy register. Old software can use this register at offset 0x24. However, new software should use the EMIFF_MRS_NEW register at 0x70.
‡ Not applicable to X or POMAP5912 silicon

Table 30. EMIFF Registers (Continued)

Base Address = 0xFFFE CC00			
Name	**Description**	**R/W**	**Offset**
EMIFF_OP	EMIFF SDRAM operation register	R/W	0x80
EMIFF_CMD	EMIFF SDRAM manual command register	R/W	0x84
EMIFF_PTOR1	EMIFF dynamic arbitration priority time-out1	R/W	0x8C
EMIFF_PTOR2	EMIFF dynamic arbitration priority time-out2	R/W	0x90
EMIFF_PTOR3	EMIFF dynamic arbitration priority time-out3	R/W	0x94
EMIFF_AADDR	EMIFF abort address register	R	0x98
EMIFF_ATYPER	EMIFF abort type register	R	0x9C
EMIFF_DLL_LRD_STAT	DLL LRD status register	R	0xBC
EMIFF_DLL_URD_CTRL	DLL URD control register (read lower byte)	R/W	0xC0
EMIFF_DLL_URD_STAT	DLL URD status register (read upper byte)	R	0xC4
EMIFF_EMRS2	EMIFF SDRAM EMRS2 register	R/W	0xC8
EMIFF_DLL_LRD_CTRL	DLL LRD control register	R/W	0xCC
EMIFF_DOUBLER‡	EMIFF TC Doubler	R/W	0x60

† EMIFF_MRS is a legacy register. Old software can use this register at offset 0x24. However, new software should use the EMIFF_MRS_NEW register at 0x70.

‡ Not applicable to X or POMAP5912 silicon

Table 31. EMIFF Enable and Disable TC Doubler Feature EMIFF_DOUBLER

Base Address = 0xFFFE CC00, Offset = 0x08				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:1	Reserved	Reserved. Read is undefined. Write must be zero.	R	0x0000
0	DOUBLER_EN	This flag indicates whether the doubler is enabled or disabled. 0: Disabled (value at reset) 1: Enabled	R/W	0

Table 32. EMIFF Priority Register (EMIFF_PRIOR)

Base Address = 0xFFFE CC00, Offset = 0x08				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:16	RESERVED	Reserved	R	0x0000
15:12	L3_OCP	L3 OCP consecutive access	R/W	0000
11:8	DMA	DSP/System DMA consecutive access	R/W	0000
7	RESERVED	Reserved	R	0
6:4	DSP	DSP core consecutive access	R/W	000
3	RESERVED	Reserved	R	0
2:0	MPU	MPU core consecutive access	R/W	000

The TC doubler reduces the latency (8 TC cycles) between the two 4 X 32 bit accesses (for 8 X 32-bit MPU access requests) by acting by acting as a fifth initiator to the EMIFF (among other initiators such as the MPU, DSP, system DMA, and OCP−I). This is applicable only to production silicon devices (not XOMAP5912 or POMAP5912 devices). The TC doubler does not double the access time between the MPU and the EMIFF. Also, the TC doubler only speeds up accesses between the MPU and EMIFF (i.e. not with the DSP, system DMA, and so forth).

The EMIFF priority register allows the EMIFF to give consecutive accesses to a master when the master has been granted the EMIFF interface. The MPU core and DSP core can have 0 to 7 consecutive accesses and the DSP/System DMA and the Level3 OCP initiator can have 0 to 15 consecutive accesses, according to the content of the corresponding bits.

Table 33. EMIFF SDRAM Configuration Register (EMIFF_CONFIG)

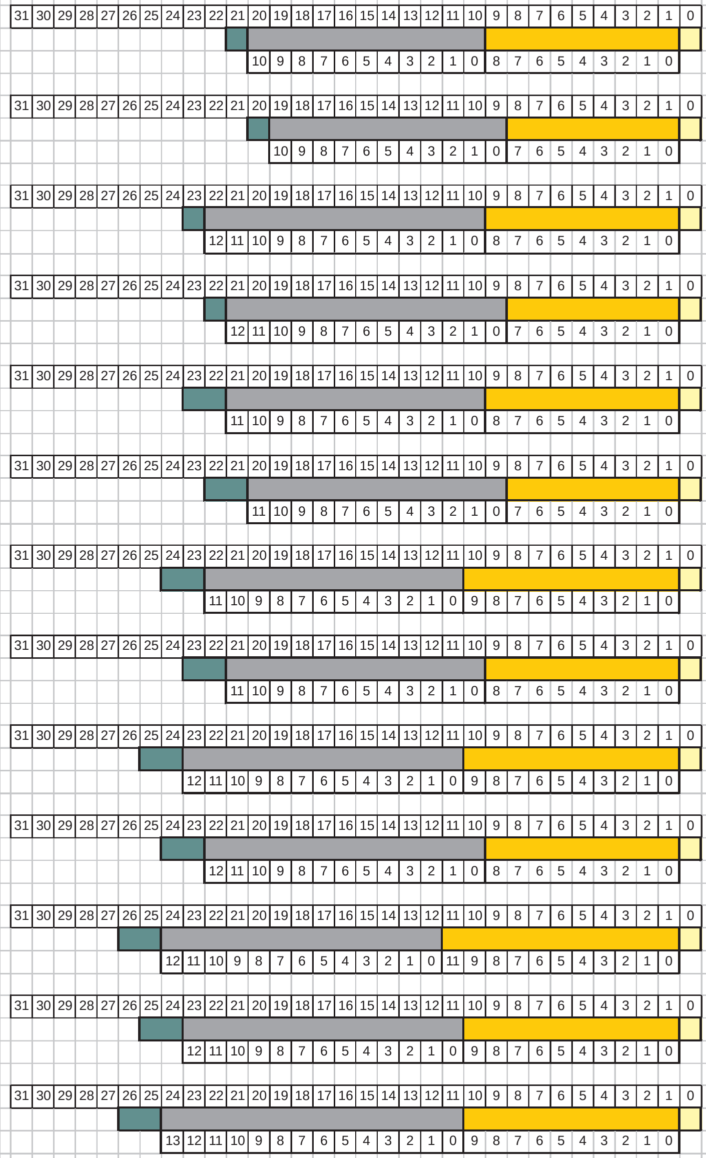
Base Address = 0xFFFE CC00, Offset = 0x20				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:30	Reserved	Must be 00	R	00
29:28	LG SDRAM Type	Used to define the larger SDRAM memories (256MB). See Table 34.	R/W	00
27	CLK	Disable SDRAM clock. 0: Enable the clock to the external SDRAM bank. 1: Disable the clock to the external SDRAM bank.	R/W	0
26	PWD	Power down enable. Puts the SDRAM device into power down mode. The CKE signal to the device is held high only for an active transaction. This bit must be enabled if the autoclock gating is used (see SD_AUTO_CLK in the SDRAM configuration 2 register, Table 56).	R/W	0

Table 33. EMIFF SDRAM Configuration Register (EMIFF_CONFIG) (Continued)

Base Address = 0xFFFE CC00, Offset = 0x20				
Bit	**Field**	**Description**	**R/W**	**Reset**
25:24	SDRAM frequency	SDRAM frequency range. To control the idle time of SDRAM regarding to the clock organization: 00: SDF0 (reset value) 01: SDF1 10: SDF2 11: SDF3	R/W	00
23:8	ARCV	Autorefresh counter register value. This value is calculated using the formula: Value = (refresh interval/clock period/number of rows) − 50	R/W	0x6188
7:4	SDRAM type	Set the SDRAM internal organization.	R/W	0x0
3:2	ARE	Autorefresh enable. When autorefresh enable is set, the EMIFF generates a REFR request, depending on the autorefresh counter and the burst refresh counter. If refresh enable is not set, the refresh must be done as a RAS_only refresh under CPU control. 00: Autorefresh disable 01: Autorefresh enable (one command every 14.7 µs) 10: Autorefresh by burst of 4 commands 11: Autorefresh by burst of 8 commands	R/W	00
1	Reserved	Must be 1.	R	1
0	Slrf	Self-refresh. When set, it puts the SDRAM in self-refresh mode if an access is made to the SDRAM after the SDRAM automatically comes out of self-refresh.	R/W	0

Table 34. EMIFF SDRAM Register Memory/Data Bus Size

SDRAM address multiplexing with respect to system 32−bit address



SDRAM Type Size/Org./Bank	Number of devices	EMIFF config register [29:28][7:4]	Memory space size
16M x 8 x 2b	2	000000	4 Mbytes
16M x 16 x 2b	1	000010	2 Mbytes
64M x 8 x 2b	2	000100	16 Mbytes
64M x 16 x 2b	1	000110	8 Mbytes
64M x 8 x 4b	2	000101	16 Mbytes
64M x 16 x 4b	1	000111	8 Mbytes
128M x 8 x 4b	2	001001	32 Mbytes
128M x 16 x 4b	1	001011	16 Mbytes
256M x 8 x 4b	2	001101	64 Mbytes
256M x 16 x 4b	1	001111	32 Mbytes
512M x 8 x 4b	2	010000	128 Mbytes
512M x 16 x 4b	1	001101	64 Mbytes
1G x 16 x 4b	1	100000	128 Mbytes

🟧	=	Column Address
▒	=	Row Address
🟩	=	Bank Address
🟨	=	Byte enable

Table 35. Frequency Range (SDRAM)

ac Parameters	SDF0 (Cycles)	SDF1 (Cycles)	SDF2 (Cycles)	SDF3 (Cycles)
t_{RC}	9	5	3	2
t_{RAS}	6	3	3	2
t_{RP}	3	2	2	2
t_{RCD}	3	2	2	2
t_{RRD}	2	2	2	2

Table 36. Frequency Range (Mobile DDR)

ac Parameters	SDF0 (Cycles)	SDF1 (Cycles)	SDF2 (Cycles)	SDF3 (Cycles)
t_{RC}	12	7	3	2
t_{RAS}	8	5	3	2
t_{RP}	5	4	2	2
t_{RCD}	4	2	2	2
t_{RRD}	3	2	2	2

Depending on the value programmed into the SDRAM_TYPE field of the EMIFF SDRAM operation register, either the ac parameters for SDRAM or the ac parameters for mobile DDR are supported; that is, the SDRAM frequency field is decoded into the values in either Table 35 or Table 36, not both. See section 3.3.7 for more information.

For production silicon only, depending on the NEW_SYS_FREQ bit in the EMIFF_CONF2 register (bit3), the ac timing is shown in Table 37.

Table 37. AC Timing When NEW_SYS_FREQ (bit 3 of EMIFF_CONFIG2) = 0

ac Parameters	SDF0	SDF1	SDF2	SDF3	SDF0	SDF1	SDF2	SDF3
SDR:EMIFF_OP[1:0] = 00					**Mobile SDR:EMIFF_OP[1:0] = 10**			
t_{RC}	9	5	3	3	12	7	3	2
t_{RAS}	6	3	3	2	8	5	3	2
t_{RCD}	3	2	2	2	4	2	2	2
t_{RP}	3	2	2	2	5	4	2	2
t_{RRD}	2	2	2	2	3	2	2	2
t_{RFC}	9	5	3	2	12	7	3	2
t_{WR}	2	2	2	2	2	2	2	2
t_{XSR}	9	5	3	2	12	7	3	2
t_{CCD}	1	1	1	1	1	1	1	1
t_{CDL}	1	1	1	1	1	1	1	1
t_{MRD}	2	2	2	2	2	2	2	2
SDR:EMIFF_OP[1:0] = 01					**Mobile SDR:EMIFF_OP[1:0] = 11**			
t_{RC}	9	5	3	3	12	7	3	3
t_{RAS}	6	3	3	2	8	5	3	2
t_{RCD}	3	3	3	3	4	3	3	3
t_{RP}	3	2	2	2	5	4	2	2
t_{RRD}	2	2	2	2	3	2	2	2
t_{RFC}	13	5	3	2	16	7	3	3
t_{WR}	3	3	3	3	3	3	3	3
t_{XSNR}	209	205	209	209	21	14	7	3
t_{CCD}	4	4	4	4	4	4	4	4
t_{CDL}	2	2	2	2	2	2	2	2
t_{MRD}	2	2	2	2	2	2	2	2

Table 38. AC Timing When NEW_SYS_FREQ (bit 3 of EMIFF_CONFIG2) = 1

ac Parameters	**SDF0**	**SDF1**	**SDF2**	**SDF3**	**SDF0**	**SDF1**	**SDF2**	**SDF3**
SDR:EMIFF_OP[1:0] = 00					**Mobile SDR:EMIFF_OP[1:0] = 10**			
t_{RC}	9	7	6	3	11	9	7	3
t_{RAS}	6	5	4	2	7	7	5	2
t_{RCD}	3	2	2	2	4	3	2	2
t_{RP}	3	2	2	2	4	3	2	2
t_{RRD}	2	2	2	2	2	2	2	2
t_{RFC}	9	7	6	3	13	10	7	4
t_{WR}	2	2	2	2	2	2	2	2
t_{XSR}	9	7	6	3	15	12	8	4
t_{CCD}	1	1	1	1	1	1	1	1
t_{CDL}	1	1	1	1	1	1	1	1
t_{MRD}	2	2	2	2	2	2	2	2
SDR:EMIFF_OP[1:0] = 01					**Mobile SDR:EMIFF_OP[1:0] = 11**			
t_{RC}	9	7	6	3	11	9	7	3
t_{RAS}	6	5	4	2	7	7	5	2
t_{RCD}	3	3	3	3	4	3	3	3
t_{RP}	3	2	2	2	4	3	2	2
t_{RRD}	2	2	2	2	2	2	2	2
t_{RFC}	10	8	6	3	13	11	8	4
t_{WR}	2	2	2	2	2	2	2	2
t_{XSNR}	200	200	200	200	27	27	14	7
t_{CCD}	4	4	4	4	4	4	4	4
t_{CDL}	0	0	0	0	0	0	0	0
t_{MRD}	2	2	2	2	2	2	2	2

Table 39. EMIFF SDRAM MRS Register (legacy for OMAP3.1)

Base Address = 0xFFFE CC00, Offset = 0x24				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:10	Reserved	Must be all 0.	R	0x000000
9	WBST	Write burst must be 0 (burst write same as burst read).	R/W	0
8:7	Reserved	Must be 00.	R	00
6:4	CASL	CAS idle time must be set to 3 for a mobile DDR device: 001: Reserved. 011: CAS idle time = 3 (default).	R/W	011
3	S/I	Serial = 0 / Interleave = 1 (must be serial).	R/W	0
2:0	PGBL	Page burst length. Must be set to full page burst (111) for SDRAM and burst of 8 (011) for DDR SDRAM.	R/W	111

Note: After the memory exits self-refresh mode, the first thing the software should do is write to the EMIFF_MRS (legacy or new) register so that the SDRAM mode register is set properly.

Table 40. EMIFF SDRAM Configuration Register 2 (EMIFF_SDRAM_CONFIG_2_REG)

Base Address = 0xFFFE CC00, Offset = 0x3C				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:4	Reserved	Must be all 0.	R	0x0000000
3	NEW_SYS_FREQ	Applicable to production silicon only. 0: The sys_freq[1:0] field of the SDRAM_CONFIG register selects the optimized AC Table. 1: Indicates system frequency is greater than 100 MHz. When set, this bit along with the sys_freq[1:0] field of the SDRAM_CONFIG register selects the optimized AC Table. When this feature is enabled, it improves the overall throughput of the memory controller	R/W	0

Table 40. EMIFF SDRAM Configuration Register 2 (EMIFF_SDRAM_CONFIG_2_REG) (Continued)

Base Address = 0xFFFE CC00, Offset = 0x3C				
Bit	**Field**	**Description**	**R/W**	**Reset**
2	sd_auto_clk	Allow controller to suspend its internal clocks when idle. The clocks are automatically reenabled when there is an autorefresh or host request. 0: Disable (Reset). 1: Enable. This bit must be set in conjunction with the CLK bit in the SDRAM configuration register in order to turn off the clock to the external SDRAM device. Also, the PWD bit in the configuration register must be set to 1 for autogating to be effective.	R/W	0
1	Rfrsh_reset	Place the SDRAM into self_refresh when in reset (active high).	R/W	1
0	Rfrsh_stdby	Place the SDRAM into self_refresh when in standby mode (active high).	R/W	1

Table 41. DLL WRD Control Register (EMIFF_DLL_WRD_CTRL)

Base Address = 0xFFFE CC00, Offset = 0x64				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:26	Reserved	Must be all 0.	R	0x00
25:20	WO	Write offset. 6-bit delay fine adjustment, signed, range –32…+31. One step represents a 26.3 ps±10.5 ps delay adjustment. Effective in both DLL enabled and DLL disabled mode. Used for delaying the write clock signal	R/W	0x00
19:16	Reserved	Must be all 0.	R	0000
15:8	DLY	Delay. 8-bit delay to adjust the digitally controlled delay, to be used when the DLL is disabled. Range 0…225 One step represents a 26.3 ps±10.5 ps delay adjustment.	R/W	0x00
7:4	Reserved	Must be all 0s.	R	0000

Table 41. DLL WRD Control Register (EMIFF_DLL_WRD_CTRL) (Continued)

Base Address = 0xFFFE CC00, Offset = 0x64				
Bit	**Field**	**Description**	**R/W**	**Reset**
3	LDLL	Load DLL. Allows loading the delay field value into the DLL module, as the initial value for the tracking counter, or to force a given delay. 0: No action. 1: The DLL is loaded with the delay value, if ENADLL is 1. The DLL tracking engine is stalled.	R/W	0
2	DLLP	DLLPhase. Nominal digitally controlled delay selection. This bit has no effect if DLL is disabled 0: 72 degrees (20% of the clock period). 1: 90 degrees (25% of the clock period).	R/W	0
1	ENADLL	0: DLL is disabled. 1: DLL is enabled.	R/W	0
0	RESERVED	Must be 0.	R	0

Table 42. DLL WRD Status Register (EMIFF_DLL_WRD_STAT)

Base Address = 0xFFFE CC00, Offset = 0x68				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:16	Reserved	Must be all 0.	R	0x0000
15:8	CNT	DLL Count. Current DLL counter value for monitoring/debug (assumes control bit ENADLL is 1 in DLL Control register.	R	0x00
7:3	Reserved	Must be all 0.	R	0x00
2	LOCK	DLL lock status (future, not in the current design) 0: DLL is not locked. 1: DLL is properly locked.	R	0
1	UDF	Underflow status 0: DLL is OK. 1: DLL counter underflow.	R	0
0	OVF	Overflow status 0: DLL is OK. 1: DLL counter overflow.	R	0

Table 43. EMIFF SDRAM MRS_NEW Register (EMIFF_MRS_NEW)

Base Address = 0xFFFE CC00, Offset = 0x70				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:10	Reserved	Must be all 0.	R	0x000000
9	WBST	Write burst must be 0 (burst write same as burst read).	R/W	0
8	ResetDLL	This bit resets the memory device DLL when set to 1.	R/W	0
7	Reserved	Must be 0.	R	0
6:4	CASL	CAS idle time: 001: Reserved 010: CAS idle time = 2 011: CAS idle time = 3 (default) Must be set to 2 for DDR	R/W	011
3	S/I	Serial = 0/Interleave=1 (must be serial).	R/W	0
2:0	PGBL	Page burst length. Must be set to full page burst (111) for SDRAM and burst of 8 (011) for DDR SDRAM.	R/W	111

Note: After the memory exits self−refresh mode, the first thing the software should do is write to the EMIFF_MRS (legacy or new) register so that the SDRAM mode register is set properly.

Note that there is only one physical MRS register. Using that address, no automatic initialization sequence is generated; only a LOAD MODE register command is issued. A CPU write to this register generates a LOAD MODE register command, with BA1,BA0 = 0,0.

Table 44. EMIFF DDR SDRAM Register (EMIFF_EMRS0)

Base Address = 0xFFFE CC00, Offset = 0x74				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:10	Reserved	Must be all 0.	R	0x0000000
9:3	Unused	No special function, but these bits are passed to the memory.	R/W	0x00
2	QFC	QFC enable bit. 0: Disabled**.** 1: Enabled.	R/W	0

Table 44. EMIFF DDR SDRAM Register (EMIFF_EMRS0) (Continued)

Base Address = 0xFFFE CC00, Offset = 0x74				
Bit	**Field**	**Description**	**R/W**	**Reset**
1	DS	DS driver strength bit. 0: Normal. 1: Reduced.	R/W	0
0	DLL	DLL enable bit. 00: Enabled. 1: Disabled.	R/W	0

This register is used for DDR SDRAM memory only. It provides access to extended configuration fields. The description is given here with reference to standard devices, but must be checked with the specification of the device actually used in a given application.

A CPU write to this register generates a LOAD MODE register command, with BA1=0 and BA0 = 1. Twelve bits can be loaded.

Table 45. EMIFF Low Power SDRAM Register (EMIFF_EMRS1)

Base Address = 0xFFFE CC00, Offset = 0x78				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:10	Reserved	Must be all 0.	R	0x000000
9:5	Unused	No special function, but these bits are passed to the memory.	R/W	0000
4:3	TCSR	Temperature-compensated self-refresh 00: 70°C maximum temperature 01: 45°C maximum temperature 10: 15°C maximum temperature 11: 85°C maximum temperature	R/W	00
2:0	PASR	Partial-array self-refresh 000: All banks 001: 1/2 array (BA1=0) 010: 1/4 array (BA1=BA0=0) 011: Reserved 100: Reserved 101: 1/8 array (BA1=BA0=0, RA11=0) 110: 1/16 array (BA1=BA0=0, RA11=RA10=0) 111: Reserved	R/W	000

Note: Bit designation is given for a 128M-bit device.

This register is used for SDRAM memories dedicated for low power wireless applications. The description is given here with reference to standard devices, but must be checked with the specification of the device actually used in a given application.

A CPU write to this register generates a LOAD MODE register command, with BA1=1 and BA0 = 0. Twelve bits can be loaded.

Table 46. EMIFF SDRAM Operation Register (EMIFF_OP)

Base Address = 0xFFFE CC00, Offset = 0x80				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:25	Time-out_B3	Time-out value for Bank3, in TC clock cycles.	R/W	0x00
24:18	Time-out_B2	Time-out value for Bank2.	R/W	0x00
17:11	Time-out_B1	Time-out value for Bank1.	R/W	0x00
10:4	Time-out_B0	Time-out value for Bank0.	R/W	0x00

Table 46. EMIFF SDRAM Operation Register (EMIFF_OP) (Continued)

Base Address = 0xFFFE CC00, Offset = 0x80				
Bit	**Field**	**Description**	**R/W**	**Reset**
3:2	Operation mode	00: Low-power/low-bandwidth mode (LPLB mode) 01: High-power/high-bandwidth mode (HPHB mode) 10: Programmable operating mode 0 (POM0 mode) 11: Reserved. Must not be used.	R/W	01
1:0	SDRAM type	The type of SDRAM: 00: Regular SDR SDRAM 01: Regular DDR SDRAM 10: Low-power SDR SDRAM 11: Mobile DDR SDRAM	R/W	00

Table 47. EMIFF SDRAM Manual Command Register (EMIFF_CMD)

Base Address = 0xFFFE CC00, Offset = 0x84				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:4	Reserved	Must be 0000:000.	R	0x0000000
3:0	SDRAM	Manual command 0000: NOP command 0001: Precharge command 0010: Autorefresh command 0011: Enter deep sleep command 0100: Exit deep sleep command 0111: Set CKE signal high 1000: Set CKE signal low 1xxx: Reserved. Not for use.	R/W	0x0

Table 48. EMIFF Dynamic Arbitration Priority Time-Out Register 1 (EMIFF_PTOR1)

Base Address = 0xFFFE CC00, Offset = 0x8C				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:7	Reserved	Must be all 0.	R	0x000000
7:0	DMA	Number of clock cycles before DSP/System DMA requests are made high priority in the dynamic priority scheme for the EMIFF SDRAM interface.	R/W	0x00

Table 49. EMIFF Dynamic Arbitration Priority Time-Out Register 2 (EMIFF_PTOR2)

Base Address = 0xFFFE CC00, Offset = 0x90				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:7	Reserved	Must be all 0.	R	0x000000
23:16	DSP	Number of clock cycles before DSP core requests are made high priority in the dynamic priority scheme for the EMIFF SDRAM interface.	R/W	0x00
15:8	Reserved	Reserved.	R	0x00
7:0	LCD	Number of clock cycles before LCD requests are made high priority in the dynamic priority scheme for the EMIFF SDRAM interface.	R/W	0x00

Table 50. EMIFF Dynamic Arbitration Priority Time-Out Register 3 (EMIFF_PTOR3)

Base Address = 0xFFFE CC00, Offset = 0x94				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:7	Reserved	Must be all 0.	R	0x000000
7:0	L3	Number of clock cycles before L3 OCP initiator requests are made high priority in the dynamic priority scheme for the EMIFF SDRAM interface.	R/W	0x00

Time-out registers 1−3 are used to store the number of clock cycles before DSP core, DSP/System DMA, or Level 3 OCP initiator requests are made high priority in the dynamic priority scheme for the EMIFF SDRAM interface.

Table 51. EMIFF Abort Address Register (EMIFF_AADDR)

Base Address = 0xFFFE CC00, Offset = 0x98				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:0	Abort Address	Address of the transaction aborted.	R	0x00000000

Table 52. EMIFF Abort Type Register (EMIFF_ATYPER)

Base Address = 0xFFFE CC00, Offset = 0x9C				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:3	Reserved	Must be all 0s.	R	0x0000000
2:1	HOSTID	ID of the host whose transaction was aborted 00: MPU core 01: DSP core 10: DMA 11: OCP-I	R	00
0	ABORT_FLAG	Set when an abort occurs, reset when this register is read.	R	0

Table 53. DLL LRD Status Register (EMIFF_DLL_LRD_STAT)

Base Address = 0xFFFE CC00, Offset = 0xBC				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:0	Reserved	Must be all 0s.	R	0x00000000

Table 54. DLL URD Control Register (EMIFF_DLL_URD_CTRL)

Base Address = 0xFFFE CC00, Offset = 0xC0				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:26	Reserved	Must be all 0s.	R	0x00
25:20	Read offset	6-bit QDS delay fine adjustment, signed, range –32…+31. Effective in both DLL enabled and DLL disabled mode. Used for delaying the read dqs for the upper byte. One step represents a 26.3 ps±10.5 ps delay adjustment.	R/W	0x00
19:16	Reserved	Must be all 0s.	R	0x0
15:8	Delay	8-bit delay to adjust the digitally controlled delay, used when the DLL is disabled. Range 0…225 One step represents a 26.3 ps±10.5 ps delay adjustment.	R/W	0x00
7:4	Reserved	Must be all 0s.	R	0x0

Table 54. DLL URD Control Register (EMIFF_DLL_URD_CTRL)(Continued)

Base Address = 0xFFFE CC00, Offset = 0xC0				
Bit	**Field**	**Description**	**R/W**	**Reset**
3	LOADDLL	Allows loading the delay field value into the DLL module, as the initial value for the tracking counter, or to force a given delay. 0: No action. 1: The DLL is loaded with the delay value, if ENADLL is 1. The DLL tracking engine is stalled.	R/W	0
2	DLLPHASE	Nominal digitally controlled delay selection. This bit has no effect if DLL is disabled. 0: 72 degrees (20% of the clock period). 1: 90 degrees (25% of the clock period).	R/W	0
1	ENADLL	0: DLL is disabled. 1: DLL is enabled.	R/W	0
0	Reserved	Must be 0.	R	0

This register controls the DLL for the upper read byte.

Table 55. DLL URD Status Register (EMIFF_DLL_URD_STAT)

Base Address = 0xFFFE CC00, Offset = 0xC4				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:16	Reserved	Must be all 0.	R	0x0000
15:8	DLLCOUNT	Current DLL counter value for monitoring/debug, assumes control bit ENADLL is 1 in DLL control register.	R	0x00
7:3	Reserved	Must be all 0.	R	0x0
2	Lock	DLL lock status (future, not in the current design) 0: DLL is not locked. 1: DLL is properly locked.	R	0
1	UDF	Underflow status 0: DLL is OK. 1: DLL counter underflow.	R	0
0	OVF	Overflow status 0: DLL is OK. 1: DLL counter overflow.	R	0

This register controls the DLL for the lower read byte.

Table 56. EMIFF SDRAM Register (EMIFF_EMRS2)

Base Address = 0xFFFE CC00, Offset = 0xC8				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:0	Reserved	Must be all 0.		

This register is provided in anticipation of future designs by memory manufacturers and must not be used by current applications.

A CPU write to this register generates a LOAD MODE register command, with BA1, BA0 = 1,1. Twelve bits can be loaded.

Table 57. DLL LRD Control Register (EMIFF_DLL_LRD_CTRL)

Base Address = 0xFFFE CC00, Offset = 0xCC				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:26	Reserved	Must be all 0s.	R/W	0x00
25:20	Read offset	6-bit QDS delay fine adjustment, signed, range –32…+31. Effective in both DLL enabled and DLL disable modes. Used for delaying the read dqs for the lower byte. One step represents a 26.3 ps±10.5 ps delay adjustment.	R/W	0x00
19:0	Reserved	Must be all 0s.	R/W	0x00000

3.5.4 OCPI Registers

Table 58 lists the OCP registers. Table 59 through Table 63 describe the register bits.

Table 58. OCP Registers

Base Address = 0xFFFE C320			
Name	**Register Description**	**R/W**	**Offset**
OCPI_AFR	OCP address fault	R	0x00
OCPI_MCFR	OCP master command fault	R	0x04
OCPI_ATYPER	OCP type of abort	R/W	0x0C
OCPI_PR	OCP protection	R/W	0x14
DYNAMIC_POWER_DOWN	OCP Dynamic Power Down	R/W	0x1C

Note: The reset value of the OCP protection register is all ones, such that all buses are protected on reset.

Table 59. OCPI Address Fault Register (OCPI_AFR)

Base Address = 0xFFFE C320, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:0	Address	Address accessed by the master that causes an abort or error.	R	0

Table 60. OCPI Master Command Fault Register (OCPI_MCFR)

Base Address = 0xFFFE C320, Offset = 0x04				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:3	Reserved	Reserved. Must be 0.	R	0
2:0	MCMD	Command that caused an abort or error.	R	0

Table 61 lists the supported commands.

Table 61. Master Command Register Supported Commands

MCmd	Transaction Type
000	Idle
001	Write
010	Read
011	ReadEX
OTHER	Not supported/not interpreted

For non-supported Mcmd encoding, OCPI translates it to IDLE.

The READEX command is for atomic transfer: Read-modify-write. This command is internally decoded to generate a lock signal for the OMAP3 traffic controller and is replaced by a simple read with a lock signal to the subtargets. The READEX command *must* be followed by a write that matches the address of the READEX, as specified in Sonic’s OCP specification. A READEX command followed by a read results in unpredictable behavior.

Table 62. OCPI Type of Abort Register (OCPI_ATYPER)

Base Address = 0xFFFE C320, Offset = 0x0C				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:4	Reserved	Reserved. Must be 0.	R	0
3	Burst Error	Burst access to the MPUI or TIPB was requested.	R/C	0
2	PROTECT	Address hit a protected area.	R/C	0
1	TRGABORT	Abort coming from the accessed target.	R/C	0
0	ADDDEC	Address decoding error (initiator sent unknown address).	R/C	0

Note: R/C is clear after read.

The abort type register is cleared after being read (R/C). In other words, reading the register once returns whatever bits correspond to the abort type. On the next read, the register returns all 0s. Also, a write to this register has no effect.

Table 63. OCPI Protection Register (OCPI_PR)

Base Address = 0xFFFE C320, Offset = 0x14				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:8	Reserved	Reserved. Must be 0.	R/W	0
7	API	Access to MPUI is prohibited.	R/W	1
6	RHEA_PUB	Access to MPU core public TIPB is prohibited.	R/W	1
5	RHEA_PRIV	Access to MPU core private TIPB is prohibited.	R/W	1
4	OCPMULT	Access to OCPT multibank is prohibited.	R/W	1
3	OCPT2	Access to OCPT2 is prohibited.	R/W	1
2	OCPT1	Access to OCPT1 is prohibited.	R/W	1
1	EMIFF	Access to EMIFF is prohibited.	R/W	1
0	EMIFS	Access to EMIFS is prohibited.	R/W	1

When a bit in the OCPI_PR register is 1, access to the corresponding bit field from the OCPI bus is protected (prohibited).

When not in secure mode, these secure mode register values are ignored. When in secure mode, EMIFS CSO and TIOB private access is not allowed, regardless of the register values. Every other target secure mode is determined by the register value.

Table 64. OCPI Dynamic Power Down (DYNAMIC_POWER_DOWN) Register

Base Address = 0xFFFE C320, Offset = 0x1C				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:1	Reserved	Reserved. Read is undefined. Write must be zero.	R	0x0000 00000
0	AUTO_GATED_CLK	The dynamic clock gating logic is software controllable. It can be disabled or enabled by software. 0: Auto clock gating is disabled. By default, the auto clock gating is disabled. By default, the reset value is 0 (OMAP32 compatibility mode). 1: Auto clock gating is enabled to save power	R/W	0

3.5.5 TC Abort Management Registers (Applicable To Production Silicon Only)

When an access to a reserved memory location occurs, the TC terminates the illegal access and generates an abort interrupt.

❑ If the faulting access was a read access, then a 0 value is read.

❑ The TC raises an interrupt sets the abort flag bit in the abort register RES_SPC_ATYPER, and the requested address in the abort address register RES_SPC_ADDR.

❑ The abort ISR should read bit[0] of the RES_SPC_ATYPE register to determine the cause of the interrupt. This bit is automatically cleared by the hardware when the register is read. The table below lists the TC abort management registers.

Table 65. TC Abort Management Registers

Base Address = 0xFFFE D500			
Name	**Description**	**R/W**	**Offset**
RES_SPC_ATYPER	Reserved space address fault type	R	0x48
RES_SPC_ADDR	Reserved space fault address	R	0x4C

Table 66. Reserved Space Fault Type Register (RES_SPC_ATYPER)

Base Address = 0xFFFE D500, Offset = 0x48				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:1	Reserved	Must be 0	R	0
0	RSVAA	Reserved address abort 0: No abort has occurred 1: Abort was caused by illegal access to memory space. Reserved address space access abort.	R/C	0

Table 67. Reserved Space Fault Address Register (RES_SPC_ADDR)

Base Address = 0xFFFE D500, Offset = 0x4C				
Bit	**Field**	**Description**	**R/W**	**Reset**
31:0	AADDR	Holds the address of the access that caused the TC abort. Valid only when RSVAA is 1.	R	0x1000000

For example, an access attempt in the range 0c9000 0000 to 0xDFFF FFFF would cause a TC abort and the offending access is stored in the RES_SPC_ADDR register.

3.6 Priority Algorithms

The traffic controller provides a choice of two priority algorithms for simultaneous requests. Arbitration is performed in each TC target port (OCP-T1, OCP-T2, EMIFF, and EMIFS).

Selection of the arbitration scheme is common to all TC ports. Depending on the OMAP device, it can be either hardwired to one of the two algorithms or programmable (outside of OMAP).

❑ Enhanced round robin with LRU

In accordance with the standard round-robin scheme, the requestor having the highest priority becomes the lowest-priority requestor after it has been granted access.

The first enhancement is that when it gets the highest priority, a requestor can keep it for a programmable number of consecutive accesses.

The number of consecutive accesses is individually programmable for the MPU core, DSP core, the system DMA, and OCP-I. See OCPT1_PRIORITY, OCPT2_PRIORITY, EMIF_SLOW_PRIORITY, and EMIF_FAST_PRIORITY registers.

In parallel with the round-robin scheme, the second enhancement is that the least recently granted requestor always has the highest priority. That means if a requestor has missed its slot because it had no request pending at that time, it keeps the highest priority for the subsequent arbitration cycles.

❑ Dynamic priority order

Most of the time, the priority order is fixed: highest priority is MPU core, then DSP core, OCP-I, and lowest priority is DSP/System DMA. A programmable time-out is attached to any host except the MPU core to ensure that it is not blocked indefinitely by higher priority hosts.

When a low-priority requestor gains the arbitration (i.e., DSP core, OCP-I, or DSP/System DMA), the associated time-out counter is loaded with the programmed value and starts decrementing.

If a counter reaches 0, the associated requestor gets the highest priority for its next request (that may be already pending). If several requestors are in this situation, the priority order is: DSP/System DMA LCD, DSP core, OCP-I, and DSP/System DMA other than LCD.

4 Clock Generation and Reset Management

4.1 Overview

The clock generation and system reset module is part of the MPU core subsystem in the OMAP 3.2 platform. This module manages the clock generation modes for the microprocessor unit (MPU core) core, the digital signal processor (DSP) core, and various other subsystems (memory interface, system DMA controller, MPU core port interface (MPUI), etc.). These clocks can be controlled by software from registers described in section 4.4. It also monitors the system reset and initiates the reset sequences for each clock domain. Finally, it controls the power-saving modes and generates wake-up controls to the processors and peripherals.

Clock generation modes include:

- Programmable clocking mode (synchronous, synchronous scalable, and mix modes)
- Programmable clock for different clock domains (MPU core, DSP core, and traffic controller (TC) clock domains)
- Programmable clock for different peripherals (internal liquid crystal display (LCD) controller and external MPU core and DSP core TIPB peripherals)
- Programmable low-frequency clocks (derived from input reference clock) to supply the internal MPU core and DSP core timers
- Fixed low-frequency clocks to supply watchdog timers for the DSP core and MPU core
- Direct memory access (DSP/System DMA) clock request mechanism (provides DSP/System DMA clock during data transfer only)
- External visibility on internally generated clocks on POCLKOUT pins

System reset includes:

- Global software reset
- Reset control for the MPU core, DSP core, and external TIPB peripherals
- System and reset status monitoring

Power-saving modes and wake-up control include:

- Programmable power-saving mode and idle mode controls for the MPU core, the DSP core, the traffic controller, and their respective subdomains

- ❑ Power control for external device reset/power on (flash memory)
- ❑ Wake-up functions initiated by interrupts (MPU core and DSP core) and DSP/System DMA requests (traffic controller and TIPB) in the idle mode
- ❑ Initiation of the wake-up sequence by external devices during the idle mode

4.2 OMAP3.2 Clock Generation

The clock domains in the OMAP 3.2 hardware engine platform are synthesized by the digital phase-locked loops (DPLL). The DPLL input clock source (CK_REF) is supplied from the ULPD.

The DPLL1 output frequencies are programmable and can be further divided down to provide clocks to the MPU core, the DSP core, and the TC domains. The MPU core domain, the DSP core domain, and the TC domain are clocked from DPLL1.

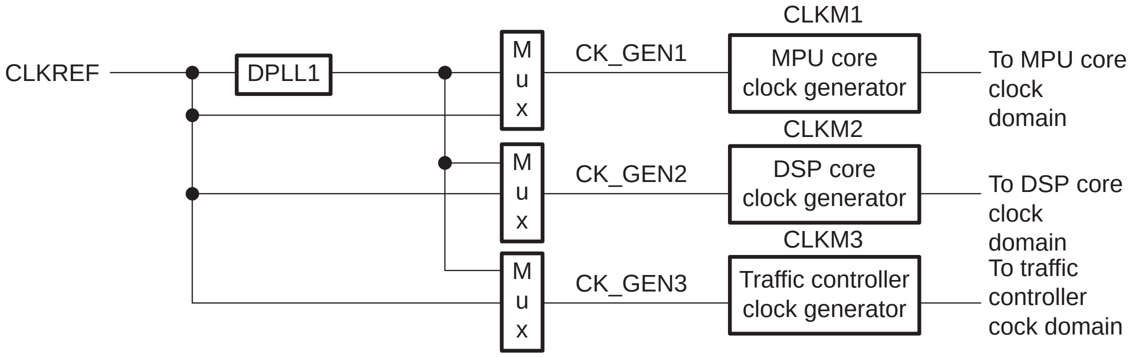
This implementation offers the clock rate selection flexibility to adjust the clock frequency of each clock domain and allows the OMAP 3.2 hardware engine to adjust each clock domain to its optimal frequency. In addition, each domain is further subdivided into subdomains, so that each subdomain can be independently activated/deactivated while the remaining part of the clock network is in an idle state.

The OMAP clock system is organized around three main clock domains: MPU core, DSP core, and TC clock domains.

- ❑ The MPU core clock domain contains: MPU core, MPU core external peripheral clocks, MPU core watchdog timer, MPU core internal timers, and MPU core interrupt handler.
- ❑ The DSP core clock domain contains: DSP core, DSP core MMU, DSP core external peripheral clocks, DSP core watchdog timer, DSP core internal timers, and DSP core interrupt handler.
- ❑ The TC clock domain contains: TC, L3 OCP-I, MPUI port interface, system DMA controller, MPU core TIPB bridges, and LCD controller, and OCP-T1 and OCP–T2.

Figure 41 shows the clock generator module.

Figure 41. Clock Generator Module



4.2.1 Clock Generation Modes

The clock generation and system reset module of the OMAP 3.2 hardware engine supports four kinds of clocking modes:

- ❑ Fully synchronous
- ❑ Synchronous scalable
- ❑ Bypass mode
- ❑ Mix modes (#3 and #4)

These clocking modes provide the system with the maximum flexibility for performance and power-saving capabilities. They are programmable by the CLOCK_SELECT field of the (ARM_SYSST) register, and the power-up default mode is the full synchronous mode.

Table 68 details the hardware engine different clocking modes.

Table 68. OMAP 3.2 Hardware Engine Clocking Modes

Clock Select	Clocking Operating Mode	MPU core Clock Source	DSP Core Clock Source	TC Clock Source	Remarks
000	Fully synchronous	DPLL1/N	DPLL1/N	DPLL1/N	See *Fully Synchronous Mode*
001	Reserved				
010	Synchronous scalable	DPLL1/M	DPLL1/N	DPLL1/O	See *Synchronous Scalable Mode*
011	Reserved				

Table 68. OMAP 3.2 Hardware Engine Clocking Modes (Continued)

Clock Select	Clocking Operating Mode	MPU core Clock Source	DSP Core Clock Source	TC Clock Source	Remarks
100	Reserved				
101	Bypass mode	CK_REF	CK_REF	CK_REF	Input reference clock
110	Mix mode #3: MPU core synchronous to TC, DSP core synchronous scalable to TC and MPU core	DPLL1/N	DPLL1/M	DPLL1/N	See *Mix Modes*
111	Mix mode #4: DSP core synchronous to TC, MPU core synchronous scalable to TC and DSP core	DPLL1/M	DPLL1/N	DPLL1/N	See *Mix Modes*

Fully Synchronous Mode

In fully synchronous mode, the MPU core, DSP core, and TC domains run at the same clock frequency derived from DPLL1. This is the power-up default mode. The fully synchronous mode is a special case of synchronous scalable mode, where the clock divider bits for all domains are equal. However, there is separate clock select encoding for fully synchronous mode. It is the programmer’s responsibility to ensure that all the clock divider select bits are set to the same value.

When the fully synchronous mode is selected, the divide-down bits of the ARM_CKCTL register must be programmed so that ARMDIV, DSPMMUDIV, DSPDIV, and TCDIV are equal.

Synchronous Scalable Mode

In synchronous scalable mode, the MPU core, DSP core, and TC domains are synchronous and run at different clock speeds. The clock feeding mechanism is similar to that of the fully synchronous mode, except that the clocks are multiples of one another.

In synchronous scalable mode, the divide-down bits ARMDIV, DSPDIV, and TCDIV of the ARM_CKCTL register define the prescaler value from the frequency of DPLL. When the synchronous scalable mode is selected, the divide-down bits of the ARM_CKCTL register must be programmed so that DSPMMUDIV = DSPDIV or DSPDIV*2.

The TC clock frequency must be the same speed or slower than the MPU core, DSP core, and the DSP core MMU clocks.

Mix Modes

Clock generation supports two mix modes.

- ❑ Mix mode #3

  The MPU core and TC clock domains are synchronous (same clock frequency), and the DSP core is scaled synchronous (synchronous but with a frequency that is a multiple of the MPU core/TC clock frequency). In this mode, the TC, MPU core, and DSP core receive clocks from the DPLL1 output.

  When mix mode #3 is selected, the divide-down bits of the ARM_CKCTL register must be programmed so that ARMDIV = TCDIV.

  The TC clock frequency must be the same speed or slower than the DSP core and the DSP core MMU clock frequencies.

- ❑ Mix mode #4

  The DSP core and TC clock domains are synchronous (same clock frequency), while MPU core is scaled synchronous (synchronous but with a frequency that is a multiple of the DSP core/TC clock frequency). In this mode, the TC, MPU core, and DSP core receive clocks from the DPLL1 output.

  When mix mode #4 is selected, the divide-down bits of the ARM_CKCTL register must be programmed so that DSPDIV = DSPMMUDIV = TCDIV. Because ARM_CK supplies the host processor and TC_CK supplies different memory interfaces, the restriction on the speed of TC_CK ensures that the rate of instruction/data fetch is never more than the rate at which data can be processed.

  The TC clock frequency must be the same speed or slower than the MPU core clock frequency.

Bypass Mode

In bypass mode (CLOCK_SELECT = 101), the DPLL is bypassed and the input reference clock is directly fed to the MPU core, DSP core, and TC clock domains.

4.2.2 DPLL

The DPLL block synthesizes a frequency clock from the fixed reference input clock signal CK_REF using the digital phase locked loop mechanism. Only the MPU core can access the DPLL control register.

DPLL Modes

The DPLL can operate either in bypass mode or in lock mode.

❑ Bypass mode

In bypass mode, the PLL_ENABLE bit of the DPLL1_CTL_REG register is set to 0. The DPLL output clock can be either CK_REF (input reference clock), CK_REF/2, or CK_REF/4, depending on the BYPASS_DIV bit-field value of the DPLL1_CTL_REG register.

❑ Lock mode

In lock mode, the PLL_ENABLE bit of the DPLL1_CTL_REG register is set to 1. The output frequency is an integer multiple or fractional multiple (m/n respectively, in the PLL_MULT and PLL_DIV bit fields of the DPLL1_CTL_REG register) of the input reference clock CK_REF. With $1 \leq m \leq 31$ and $1 \leq n \leq 4$, the frequency output ranges from CK_REF/4 to 31×CK_REF.

Synthesizing a Clock

At reset, the DPLL is in bypass mode and the BYPASS_DIV bit field of the DPLL1_CTL_REG register is set to 0b00 (DPLL output clock = CK_REF).

The procedure to synthesize a clock at a desired frequency is as follows:

1) Set the PLL_MULT and PLL_DIV bit fields of the DPLL1_CTL_REG register to the correct value in order to set the desired multiplication factor.
2) Set the PLL_ENABLE bit to 1 to enter the lock mode.
3) When the DPLL has reached the desired synthesized clock frequency, the bit LOCK bit of DPLL1_CTL_REG register goes to 1 and the output clock receives the synthesized clock.

The bit fields PLL_MULT and PLL_DIV can be modified on-the-fly, even when the DPLL is in lock mode.

Polling can be done on the LOCK bit to determine when the DPLL locks on the desired synthesized frequency. The DPLL output clock is switched smoothly between the bypass and the locked frequency because it is not mandatory to wait for the DPLL to enter lock mode before proceeding the DPLL output clock.

When idle mode of the DPLL is exited, the DPLL is set in bypass mode and the output signal is valid (locked) after a maximum of 10 input reference clock cycles. The output is valid after a maximum of 12 input reference clock cycles in bypass mode and switches to locked clock in another 32 maximum reference clock cycles. If the DPLL was synthesizing a frequency prior to the idle state, the DPLL switches from bypass mode to synthesizer frequency when the lock state is reacquired.

4.2.3 MPU Core Clock Domain

The DPLL1 output frequency defines the speed of the MPU core and the MPU core external peripherals. The clock from DPLL1 output is supplied to the OMAP boundary. It has a software gating in the ARM_IDLECT1 (IDL_CLKOUT_ARM) and in the ARM_IDLECT2 (EN_CKOUT_ARM).

At reset, DPLL1 is in bypass mode (CK_GEN1 = CK_REF).

The MPU core clock domain is subdivided into five subdomains.

- ❑ MPU core (ARM_CK)

  The divide-down ARMDIV bits of the ARM_CKCTL register can be programmed to have the DPLL1 output clock (CK_GEN1) further divided by 1, 2, 4, or 8 to supply the clock signal driving the MPU core. At reset, the highest frequency (divided by 1) is selected: ARM_CK = CK_GEN1 = CK_REF.

- ❑ MPU core external peripheral (ARMPER_CK or ARMXOR_CK)

  The divide-down PERDIV bits of the ARM_CKCTL register can be programmed to have CK_GEN1 further divided by 1, 2, 4, or 8 to supply the MPU core external peripheral clock ARMPER_CK signal at the OMAP boundary. At reset, the highest frequency (divided by 1) is selected and ARMPER_CK is active. ARMXOR_CK, a gated version of CK_REF, can also be used to supply the external peripherals. At reset, this clock is inactive.

- ❑ OMAP3.2 MPU core internal OS timers (ARMTIM_CK)

  The ARM_TIMXO bit of the ARM_CKCTL register selects either CK_GEN1 divided by 1, or the input reference clock (CK_REF) to supply the internal MPU core timers. At reset, CK_GEN1 is selected but the timer clock is inactive.

- ❑ MPU core Level 1 and 2 interrupt handlers (ARM_INTH_CK)

  The MPU core interrupt handlers are supplied by a programmable clock, and the user can choose between the MPU core clock or the divided-by-2 MPU core clock using the ARM_INTHCK_SEL bit of the ARM_CKCTL register. The MPU core clock is supplied as the default clock.

- ❑ 32-bit MPU core watchdog timer (ARMWDI_CK)

  The 32-bit MPU core watchdog timer is supplied with a low-frequency clock (CK_REF/14). This clock is active at reset.

Even when the MPU core is not in idle mode, there is the option of individually disabling the clock to the MPU core subdomains via the ARM_IDLECT2 register. This allows power saving when a module is not used.

4.2.4 DSP Core Clock Domain

Depending on the OMAP clocking mode, the DPLL1 output frequency defines the speed of the DSP core domain. The clock output from DPLL1 (CK_GEN2) can be further divided in order to supply the clock of the DSP core and its subsystems. At reset, DPLL1 is in bypass mode (CK_GEN1 = CK_GEN2 = CK_REF).

The DSP core clock domain is divided into six subdomains:

❑ DSP core (DSP_CK)

The clock signal driving the DSP core can be further divided by 2, 4, or 8 by programming the divide-down bits DSPDIV of the ARM_CKCTL register. At reset, the highest frequency (divided by 1) is selected and DSP_CK = CK_GEN2 = CK_REF.

❑ DSP core MMU (DSPMMU_CK)

The clock signal driving the DSP core MMU (DSPMMU_CK) can be further divided by 2, 4, or 8 by programming the divide-down bits DSPMMUDIV of the ARM_CKCTL register. At reset, the highest frequency (divided by 1) is selected, but the DSP core MMU clock is inactive.

Note:

DSPDIV and DSPMMUDIV must be programmed so that the DSPMMU_CK clock frequency is either one or one-half times the DSP_CK clock frequency.

❑ DSP core external peripheral clock (DSPPER_CK)

The DSP core external peripheral clock can be further divided by 2, 4, or 8 by programming the divide-down bits PERDIV of the DSP_CKCTL register.

DSPXOR_CK, a gated version of the CK_REF, can also be used to supply the external peripherals. This clock is inactive at reset.

❑ DSP core watchdog timer (DSPWDT_CK)

DSP core watchdog timer is supplied with a low-frequency clock (CK_REF/14). This clock is active at reset.

❑ DSP core internal timers (DSPTIM_CK)

The TIMXO bit of the DSP_CKCTL register selects either CK_GEN2 divided by 2, or the input reference clock (CK_REF) to supply the internal DSP core timers. At reset, the clock issued from the DPLL is selected but the timer clocks are inactive.

- ❑ DSP core Level 1 and 2.0 interrupt handlers (DSP_INTH_CK)

  DSP core interrupt handlers are supplied with CK_GEN2 divided by 2.

Even when the DSP core is not in idle mode, there is the option of individually disabling these subdomains using the DSP_IDLECT2 register. This allows significant power saving when a module is not in use.

4.2.5 Traffic Controller Clock Domain

The DPLL output frequency, which drives the traffic controller, generates the traffic controller clock (TC_CK). This TC_CK feeds the traffic controller, the OCP initiator port (OCP-I), the OCP Target1 (OCP-T1) and OCP Target2 (OCP-T2) ports, the system DMA controller, the LCD controller, the MPUI port interface, and the MPU core TIPB bridge. TC1_CK and TC2_CK are then broadcast outside the OMAP platform; they are identical to TC_CK and can be powered down independently in power saving options.

The TC clock domain is divided into two subdomains:

- ❑ Traffic controller, OCP-I port, OCP-T1 and OCP-T2 ports, MPUI port interface, system DMA controller, and MPU core TIPB bridges.

  The clock signal driving these modules is basically the same as the TC_CK, except that it can be gated independently of TC_CK.

  The divide-down TCDIV bits of the ARM_CKCTL register can be programmed to have the CK_GEN3 further divided by 2, 4, or 8 to generate the TC_CK. At reset, the highest frequency (divided by 1) is selected and TC_CK = CK_REF.

  At reset, the MPUI port interface clock and the system DMA controller clock are inactive, while the OCP-I, OCP-T1, and OCP-T2 port clocks, TC clocks, and MPU core TIPB bridge are active.

- ❑ LCD controller

  The divide-down LCDDIV bits of the ARM_CKCTL register can be programmed to have CK_GEN3 further divided by 2, 4, or 8 to generate the LCD controller clock. At reset, the highest frequency (divided by 1) is selected but the LCD controller clock is inactive.

These traffic controller subdomain clocks can be disabled even if the MPU core, DSP core, or TC are not in idle mode using the ARM_IDLECT2 register.

The DSP/System DMA needs a free-running clock supplied to the external LCD controller even when the DSP/System DMA clock is turned off so that the LCD controller can generate the proper interrupts. This free-running clock for external LCD controller can only be cut off when the external LCD controller is in idle state.

4.3 Power-Saving Modes and Wake-Up Control

This section describes the following power-saving features:

- ❑ MPU core idle control
- ❑ DSP core idle control
- ❑ Traffic controller idle control
- ❑ System DMA idle control
- ❑ MPU core TIPB bridge idle control
- ❑ External device power control
- ❑ DPLL idle control
- ❑ Chip idle mode/deep sleep mode
- ❑ Wake-up control

4.3.1 MPU Core Idle Control

The OMAP 3.2 hardware engine operates in several power-saving modes that reduce the operating current by stopping the clock signals of unused (inactive) domains, without losing any data on operational context. When the idle state is entered, the MPU core domain clocks are turned off according to the sequenced events. The clock gating cell design ensures that clocks are properly stopped and restarted without parasitic pulses.

Activating the wait-for-interrupt MPU core instruction initiates the MPU core idle mode. It stops the MPU core internal clocks, and then the STANDBYWFI signal from MPU core megacell is asserted high, indicating that the MPU core internal idle state is entered.

Before the idle mode is entered, the MPU core internal timer clock, the LCD clock, the external peripheral clock, and the timer/watchdog clock can be stopped by setting the corresponding bits of the ARM_IDLECT2 register to 0, or the corresponding bits of ARM_IDLECT1 register can be set so that these peripherals automatically go to idle when the MPU core goes to idle (except for the LCD).

When the timer/watchdog timer is configured as a watchdog, its clock (CK_REF/14) is never shut down, regardless of the value of the IDLWDT_ARM bit in the ARM_IDLECT1 register, or the EN_WDTCK bit in the ARM_IDLECT2 register.

The idle command is forwarded to the MPU core interrupt handler and the MPU core clock is stopped when the interrupt handler acknowledges this request (no pending interrupts).

When the MPU core internal clocks are stopped, the MPU core domain clocks are stopped, if they were not already disabled using ARM_IDLECT2 before MPU core went to idle.

- The ARM_CK is stopped in a low static state after some synchronization cycles.
- If the IDLCLKOUT_ARM bit field of ARM_IDLECT1 is set to 1, then the DPLL output clock also goes to idle after some synchronization cycles.
- If the IDLPER_ARM bit field of ARM_IDLECT1 is set to 1, then the ARMPER_CK also goes to idle if the ARMPER_IDLE_REQ request is acknowledged.
- If the IDLXOR_ARM bit field of ARM_IDLECT1 is set to 1, then the ARMXOR_CK also goes to idle after some synchronization cycles.
- If the IDLWDT_ARM bit field of ARM_IDLECT1 is set to 1, then the ARMWDT_CK also goes to idle after some synchronization cycles, if the MPU core watchdog module is not set.
- If the IDLTIM_ARM bit field of ARM_IDLECT1 is set to 1, then the ARMTIM_CK also goes to idle after some synchronization cycles.
- If the IDLAPI_ARM bit field of ARM_IDLECT1 is set to 1, then the MPUI clock also goes to idle whenever the MPUI is not required.
- If the IDLDPLL_ARM bit field of ARM_IDLECT1 is set to 1, then the DPLL macro goes to idle when all the clock domains are in idle.

Even when the MPU core is not in idle mode, the user has the option of individually disabling the MPU core subdomain clocks by setting the corresponding bit in the ARM_IDLECT2.

A wake-up sequence is initiated in the MPU core domain only upon:

- A system reset.

or

- An unmasked MPU core interrupt request, assuming that the WKUP_MODE bit of ARM_IDLECT1 is set to 1 or the chip is not in idle. If the chip is idle and WKUP_MODE is set to 0, the external wake-up control feature is enabled and a CHIP_nWKUP low is needed to wake-up the MPU core, along with the interrupt.

On wake-up, all the subdomains that were put in idle mode using ARM_IDLECT1 are restarted, if the corresponding enable bits of ARM_IDLECT2 are set.

4.3.2 DSP Core Idle Control

The DSP core idle instruction must be executed in host-only mode (HOM) to initiate the DSP core idle mode. Depending on the settings of the DSP core idle control registers (DSP_IDLECT1 and DSP_IDLECT2), different parts of the DSP core subsystem go to idle mode when the IDLE instruction is executed.

The following procedure describes how to program the DSP core to enter idle mode:

1) Disable the watchdog timer.

   When the timer/watchdog timer is configured as a watchdog, its clock (CK_REF/14) is never shut down.

2) Disable the following DSP core peripheral clocks by setting the corresponding bits in DSP_IDLECT2 register to 0s.

   - DSP core external peripheral clock
   - External reference peripheral clock

   This disables the clocks immediately, regardless of whether the DSP core clock is enabled or not. Alternatively, set the DSP_IDLECT1 corresponding bits to 1s, which disables the DSP core peripheral clocks only when the DSP core clock is disabled.

3) Switch the DSP core TIPB and MPUI to shared access mode (SAM).

4) Program the DSP core idle control register to put all the DSP core subsystem domains in idle mode.

5) Switch the DSP core TIPB and MPUI to host-only mode.

6) Execute the IDLE instruction.

7) When IDLE_DSP = 1 in ARM_SYSST register, the DSP_CK stops.

8) A signal is sent to the OMAP DSP core interrupt handler to disable the interrupts to the DSP core while the DSP core clock is being disabled.

9) The DSP core interrupt handler clock also stops after the synchronization cycles end.

10) The DSP core clock subdomain goes to idle mode when both the DSP core and the MPU core clocks are disabled and the corresponding DSP_IDLECT1 bits are set to 1.

Even when the DSP core or MPU core is not in idle mode, there is an option of individually disabling the DSP core domain clocks by setting the corresponding DSP_IDLECT2 enable bits to 0s.

A wake-up sequence is initiated in DSP core domain only upon one of the following events:

- A system reset.

or

- A DSP core reset.

or

- An unmasked DSP core interrupt request. The interrupt request restarts the DSP core clock if the WKUP_MODE bit of ARM_IDLECT1 is set to 1 or if the chip is not in idle. If the chip is idle and WKUP_MODE is set to 0, the external wake-up control feature is enabled and a CHIP_nWKUP low is required to wake up the DSP core clock, in conjunction with the interrupt.

On wake up, all of the DSP core subdomains that were put in idle mode using DSP_IDLECT1 are restarted following the sequence below, if the corresponding enable bits of ARM_IDLECT2 are set.

- Service and clear the interrupt event.
- Switch the DSP core TIPB and MPUI to SAM mode.
- Write 0 in the DSP core ICR idle mode configuration register bits for each subdomain to be restarted.
- Execute an idle instruction to force the reread of the DSP core ICR idle mode configuration register and restart the affected subdomain clocks.

4.3.3 Traffic Controller, System DMA Controller, and MPU Core TIPB Bridges Idle Control

Specific conditions must be met for the traffic controller, the system DMA controller, and the MPU core TIPB bridges to enter the idle mode.

Traffic Controller Idle Control

To enter idle mode, the traffic controller must meet the following conditions:

- ❑ MPU core and DSP core must be set to global idle mode.
- ❑ L3 OCP-I is in idle mode. This can be done either by clearing the EN_OCPI_CK bit of the ARM_IDLECT3 register to 0, or by setting the IDLOCPI_ARM bit in the same register to 1 (which allows disabling of the OCPI clock in conjunction with the MPU core clock). The OCP initiator bus enable signal (L3_OCPI_EN) must always be 0 before the OCPI clock stops.

❑ OCP-T1/T2 modules are in idle mode. This can be done by setting ARM_IDLECT3 (IDLTC1_CK) = 1 and ARM_IDLECT3 (IDLTC2_CK) = 1, which disables the TC1_CK and TC2_CK clocks when there is no activity and the idle request is acknowledged by the target. These modules can also be placed in idle mode by by setting EN_TC1_CK and EN_TC2_CK to 0, which completely disables TC1_CK and TC2_CK.

❑ The idle interface (IDLIF_ARM) bit of the ARM_IDLECT1 register is set to logical 1.

❑ The MPUI clock is idle.

❑ There are no system DMA pending transfers.

❑ All the TC subdomains are stopped by setting the appropriate bits in the ARM_IDLECT2 and ARM_IDLECT3 registers.

❑ There are no MPU core or DSP core interrupt requests.

❑ Power-down enable bits PDE and PWD_EN of the EMIFS configuration register EMIFS_CONFIG are set to logical 1.

❑ Disable the SDRAM clock and set the power-down enable bit to 1 with the EMIFF configuration register. TC idle mode entry is affected by the RFRSH_STBY bit of the EMIFF_SDRAM_CONFIG_2REG TC register. When set to 1, the SDRAM must be put into self-refresh state before going idle (SLRF bit of EMIFF_CONFIG register). Every time the SDRAM wakes up, the self-refresh state is cleared, so returning to idle requires that the SLRF bit of EMIFF_CONFIG_REG is set to 1 again.

Note that certain bits in the ARM_IDLECT1,2,3 registers can prevent the chip from entering the idle state.

- If the ARM_IDLECT2.EN_APICK bit is 1, either the ARM_EWUPCT.REPWR_EN bit needs to be 0, or the ARM_IDLECT3.IDLOCPI_ARM bit needs to be 1 to go to idle.
- If ARM_IDLECT2.EN_APICK is 0, ARM_EWUPCT.REPWR_EN and ARM_IDLECT3.IDLOCPI_ARM are don’t care, and will not prevent the chip from going to idle.
- If the EN_API_CK bit field of ARM_IDLECT2 is 1, then IDLAPI_ARM bit field of ARM_IDLECT1 needs to be 1 to go to idle.

Then the traffic controller completes its current operations and pulls the TCIDLE_ACK signal to a high level to indicate that TC_CK can now be safely stopped. In addition, the shut-down of the TC_CK clock informs the chip idle control logic to initiate the DPLL idle.

The TC_CK restarts upon:

- An MPU core or DSP core interrupt request
- A DSP/System DMA request
- L3_OCPI_EN pin set to logic 1 (enables restarting of the clock to L3 OCP initiator bus)

System DMA Idle Control

The system DMA employs a built-in power-saving mechanism. The clock is only requested to the clock generator when DMA transfers are occurring.

The DMA clock can enter idle mode if one of the following conditions is true:

- DMACK_REQ = 1, and there are no DMA requests.
- DMACK_REQ = 0, the MPU core clock is in idle, IDLIF_ARM = 1, there are no DMA requests, and the DMAIDLE_ACK signal is high.

MPU Core TIPB Bridges Idle Control

The TIPB bridges can enter idle mode only when all of the following conditions are true:

- MPU core is set in idle mode.
- The idle interface bit IDLIF_ARM of the ARM_IDLECT1 register is set to logical 1.
- There are no system DMA requests to the TIPB.
- There is no posted write (that is, posted write buffers are empty).

4.3.4 External Device Power Control

The FLASH.RP signal is an output pin that allows the reset/power-on control sequences of external devices such as flash memory. Whenever the traffic controller enters the idle mode, the FLASH.RP pin switches from a high to a low level, allowing external components to be turned off. When a wake-up condition is detected, the pin is switched back to a high level and restores power to external devices.

Setting the bit REPWR_EN of ARM_EWUPCT to logical 0 enables this capability. At reset, this is disabled.

To allow the external device/component voltage to stabilize (ramp-up) when the power-down mode is released, the external power control is implemented with a programmable counter that delays the restart of all clocks from FLASH.RP signal going high. The EXTPWR bit field of the ARM_EWUPCT register permits the delay to be defined as follows:

$WT_{\text{(wake-up time)}} = (EXTPWR_{\text{(field value)}} \pm 1) \text{ x } CK\_REF_{\text{(period)}}$

4.3.5 DPLL Idle Control

The DPLL can be set to idle mode if only the input reference clock (CK_REF) is needed.

The DPLL idle mode is entered when the IDLDPLL_ARM bit of the ARM_IDLECT1 register is set to logical 1 and all of the domains that use the DPLL clock are stopped. This means that the only domains running (domains that use CK_REF rather than the DPLL clock) are:

- MPU core and DSP core watchdog timers (CK_REF/14).
- Internal MPU core timers when the ARM_TIMXO bit of the ARM_CKCTL register is set to logical 0.
- Internal DSP core timers when the TIMXO bit of the DSP_CKCTL register is set to logical 0.

The DPLL idle mode entry/exit time can be significant. The input reference clock must be active for at least 24 input clock cycles from the idle request (idle rising edge) before the idle setup is complete. Once the idle mode is exited, the DPLL is set in bypass mode and the output signal is valid after a maximum of 10 input reference clock cycles. The total DPLL idle entry-to-exit sequence takes no less than 34 reference clock cycles. Therefore, it may be preferable not to shutdown the DPLL when MPU core or DSP core have to be stopped for a short period of time or when critical operations are likely to occur (that is, DSP/System DMA transfer, interrupt handling).

4.3.6 Chip Idle Mode, Deep Sleep Mode, and Wake-up Control

The OMAP 3.2 hardware engine is considered to be in chip idle mode when the MPU core, DSP core, peripherals, DPLL, and peripherals using CK_REF as their source are stopped.

Once the procedures for MPU core idle, DSP core idle, traffic controller idle, and DPLL idle have been followed, the chip idle state is reached. The following ordering is recommended to ensure that all of the clock domains can be made idle:

- The first domain to idle is the DSP core clock domain. Ensure that the MPUI interface has been correctly initialized, and that the API_SIZE value is zero. Then follow the sequence for DSP core idle already described: sending a command from the MPU core to DSP core via the mailbox is one way to start this sequence. If the DSP core has not been enabled after reset, it is only necessary to clear the ARM_CKCTL(EN_DSPCK) register bit to 0 in order to reach the idle state for this domain.

- The MPU core must wait until the DSP core has reached the idle state.
- Disable the MPU core watchdog timer.
- Set the ARM_IDLECT1, ARM_IDLECT2, and ARM_IDLECT3 register bits in preparation for going to idle (MPU core, DSP core, TC, and DPLL idle entries are all affected by these registers).
- Configure the EMIFS and EMIFF modules as described in the traffic controller idle control section. If SDRAM contents must be maintained during the idle state, then the SDRAM self-refresh mode must be enabled before going to chip idle. Set the SLFR bit of the EMIFF_SDRAM_CONFIG register to 1, and the RFRSH_STDBY bit of the EMIFF_SDRAM_CONFIG2 register to 1. The self-refresh bit is cleared every time OMAP leaves chip idle.
- Prepare for wake up by enabling and unmasking MPU core interrupts.
- Ensure that all interrupts and DSP/System DMA status bits have been cleared. If all of the other idle conditions and controls have been met, as per the MPU core, DSP core, TC and DPLL descriptions, then activating the wait-for-interrupt instruction at this point leads to the full chip idle state, allowing the reference clock to be stopped.

In chip idle mode, the MPU core, the DSP core, the DPLL and peripherals that use CK_REF as their source are stopped, while the external clock source remains the only active clock signal.

In deep sleep mode, all internal system clocks (MPU core, DSP core, DPLL, peripherals, and timers) and the external reference clock source are stopped, leaving the OMAP3 in a static state in which it consumes the lowest possible power. In this mode, it is recommended that the WKUP_MODE bit of the ARM_IDLECT1 register be set to 0 before going into IDLE. A complete handshake between the OMAP and external module turns off the OMAP input clock. This handshake ensures that OMAP wakes up properly.

Any unmasked interrupt request (either to the MPU core or the DSP core), any DSP/System DMA clock request, or setting the L3_OCPI_EN signal to high exits the idle mode.

When the WKUP_MODE bit of ARM_IDLECT1 is set to logical 0, the wake-up procedure can be controlled by the ULPD.

When the WKUP_MODE bit value is set to logic 1, a single wake-up condition initiates a chip wake-up procedure. The wake-up condition can be caused by:

❑ An interrupt request from the MPU core interrupt handler. The MPU core interrupt handler sets the nIRQ_SET signal to logic low and initiates the restarting of the ARM_CK, ARM_INTH_CK, Rhea_CK, DMA_CK, and TC_CK clocks. Depending on the setting of the ARM_IDLECT1/2 registers, peripherals clocks can also restart. This is a valid wake-up condition for MPU core, TC, and DPLL.

❑ An interrupt request from the DSP core level2 interrupt handler. This initiates the restarting of the DSP_CK, DSP_INTH_CK, and TC_CK clocks. Depending on the setting of the ARM_IDLECT1/2 registers, peripheral clocks can also restart. This signal must remain active until the DSP core asserts the DSP_IDLE signal low. This is a valid wake-up condition for DSP core, TC, and DPLL.

❑ L3_OCPI_EN pin: When the L3_OCPI_EN pin is pulled high, the TC_CK, TC1_CK, and TC2_CK, and the L3_OCPI_CK restart and the TC_CK, L3_OCPI_CK, TC1_CK, and TC2_CK keep running as long as the pin remains asserted high. This is a valid wake-up condition for TC and DPLL only.

❑ TCLB_DMAREQ: When the system DMA controller receives an asynchronous request from the traffic controller, this signal is set high to enable the DMA_CK/TC_CK and DMA_CK/TC_CK to keep running as long as the system DMA operates. This is a valid wake-up condition for TC and DPLL only.

❑ Rhea_DMAREQ: When the system DMA controller receives a request from the TIPB-bridge, this signal is asserted high to enable the TC_CK/Rhea_CK/DMA_CK and the TC_CK/Rhea_CK/DMA_CK to keep running as long as the system DMA operates. This is a valid wake-up condition for TC and DPLL only.

4.4 Registers

All registers are 32-bit registers. MPU core clock generation and system reset control registers are accessed by the MPU core only. DSP core control registers are accessed by the DSP core and the MPU core through the MPU core interface.

4.4.1 MPU Core Registers

The MPU core registers are listed in Table 69. Table 70 through Table 79 provide register bit descriptions.

Table 69. MPU Core Registers

Base Address = 0xFFFE CE00			
Name	**Description**	**R/W**	**Offset**
ARM_CKCTL	MPU core clock control prescaler selection	R/W	0x00
ARM_IDLECT1	MPU core idle enable control 1	R/W	0x04
ARM_IDLECT2	MPU core idle enable control 2	R/W	0x08
ARM_EWUPCT	MPU core restore power delay	R/W	0x0C
ARM_RSTCT1	Master software reset	R/W	0x10
ARM_RSTCT2	Peripherals reset	R/W	0x14
ARM_SYSST	MPU core clock reset status	R/W	0x18
ARM_CKOUT1	MPU core clock out definition	R/W	0x1C
ARM_CKOUT2	MPU core reserved	R/W	0x20
ARM_IDLECT3	MPU core idle enable control 3	R/W	0x24

Table 70. MPU Core Clock Control Prescaler Selection Register (ARM_CKCTL)

Base Address = 0xFFFE CE00, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:15	RESERVED	See note.	R/W	0x0000
14	ARM_INTHCK_SEL	This bit controls which clock is used for the ARM_INTH_CK 0: ARM_INTH_CK clock is same as ARM_CK (default). 1: ARM_INTH_CK is half the frequency of ARM_CK.	R/W	0
13	EN_DSPCK	Turns on DSP_CK while the DSP core is still in reset state 0: Disables DSP_CK activation during the reset state. 1: Enables DSP_CK activation during the reset state.	R/W	1

Note: For reserved bits, reading gives undefined values. Writing to has no effect.

Table 70. MPU Core Clock Control Prescaler Selection Register (ARM_CKCTL) (Continued)

Base Address = 0xFFFE CE00, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
12	ARM_TIMXO	Selects a subfrequency issued either from CK_GEN1, or from input reference clock (CK_REF) to supply internal MPU core timers. 0: ARMTIM_CK clock frequency is the input reference clock (CK_REF). 1: ARMTIM_CK clock frequency is issued from CK_GEN1.	R/W	1
11:10	DSPMMUDIV	Defines the prescaler value from the frequency of CK_GEN2 to DSPMMU clock domain. 00: CK_GEN2 01: CK_GEN2/2 10: CK_GEN2/4 11: CK_GEN2/8	R/W	00
9:8	TCDIV	Defines the prescaler value from the frequency of CK_GEN3 to TC clock domain 00: CK_GEN3 01: CK_GEN3/2 10: CK_GEN3/4 11: CK_GEN3/8	R/W	00
7:6	DSPDIV	Defines the prescaler value from the frequency of CK_GEN2 to DSP core clock domain 00: CK_GEN2 01: CK_GEN2/2 10: CK_GEN2/4 11: CK_GEN2/8	R/W	00
5:4	ARMDIV	Defines the prescaler value from the frequency of CK_GEN1 to MPU core clock domain 00: CK_GEN1 01: CK_GEN1/2 10: CK_GEN1/4 11: CK_GEN1/8	R/W	00

Note: For reserved bits, reading gives undefined values. Writing to has no effect.

Table 70. MPU Core Clock Control Prescaler Selection Register (ARM_CKCTL) (Continued)

Base Address = 0xFFFE CE00, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
3:2	LCDDIV	Defines the prescaler value from the frequency of CK_GEN3 to LCD controller clock signal 00: CK_GEN3 01: CK_GEN3/2 10: CK_GEN3/4 11: CK_GEN3/8	R/W	00
1:0	ARM_PERDIV	Defines the prescaler value from the frequency of CK_GEN1 to MPU core external peripheral clock domain 00: CK_GEN1 01: CK_GEN1/2 10: CK_GEN1/4 11: CK_GEN1/8	R/W	00

Note: For reserved bits, reading gives undefined values. Writing to has no effect.

Table 71. MPU Core Idle Enable Control Register 1 (ARM_IDLECT1)

Base Address = 0xFFFE CE00, Offset = 0x04				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:13	RESERVED	See note.	R/W	0x0000
12	IDL_CLKOUT_ARM	This read-write bit selects the idle entry mode for the external DPLL output clock. 0: The clock supplied to the external DPLL output clock remains active when the MPU core enters the idle mode (ARM_CK stopped). 1: The clock supplied to the external DPLL O/P clock is stopped in conjunction with the MPU core clock when the MPU core enters the idle mode (ARM_CK stopped).	R/W	0
11	RESERVED	See note.	R/W	0

Note: For reserved bits, reading gives undefined values. Writing to has no effect.

Table 71. MPU Core Idle Enable Control Register 1 (ARM_IDLECT1) (Continued)

Base Address = 0xFFFE CE00, Offset = 0x04

Bit	Name	Function	R/W	Reset
10	WKUP_MODE	Controls how the MPU core can exit the CHIP_IDLE state 0: After the interrupt has been asserted, the MPU core idle mode is exited upon a low level at the external CHIP_nWKUP pin. Also, any of the wake-up conditions only wake up the OMAP out of CHIP_IDLE if CHIP_nWKUP is low. 1: Idle mode is exited upon an MPU core interrupt (regardless the CHIP_nWKUP pin). Also, any wake-up condition wakes up the OMAP out of CHIP_IDLE regardless of the value on the CHIP_nWKUP pin.	R/W	1
9	IDLTIM_ARM	Selects the idle entry mode for internal MPU core timer clock. 0: The clock supplied to the timers remains active when the MPU core enters the idle mode. 1: The timer clock is stopped in conjunction with the MPU core clock when the idle mode is entered.	R/W	0
8	IDLAPI_ARM	Selects the idle entry mode for MPUI clock. 0: The clock supplied to the MPUI is fully controlled by EN_APICK bit. 1: The clock supplied to MPUI is on whenever it is required for any functionality; else it goes to IDLE. This bit must be set to 1 active to go to chip idle mode. The EN_APICK bit must not be deactivated to go to chip idle as that may cause wake-up problems for certain sources.	R/W	0
7	IDLDPLL_ARM	Enables the DPLL macro to enter idle mode when DSP core is set to global_idle mode, MPU core is in idle mode, no active DSP/System DMA transaction or TCLB_EN pin is asserted low, no TIPB posted write is queued, and the peripheral clocks are stopped. 0: DPLL remains active when the above conditions occur. 1: DPLL enters idle mode when above conditions are met.	R/W	0

Note: For reserved bits, reading gives undefined values. Writing to has no effect.

Table 71. MPU Core Idle Enable Control Register 1 (ARM_IDLECT1) (Continued)

Base Address = 0xFFFE CE00, Offset = 0x04				
Bit	**Name**	**Function**	**R/W**	**Reset**
6	IDLIF_ARM	Enables the TIPB bridge, the system DMA controller, and the TC to enter idle mode when the MPU core processor executes the wait-for-interrupt instruction. 0: The clocks remain active when the MPU core enters the idle mode. 1: The clocks are stopped in conjunction with the MPU core clock when idle mode is entered and the DSP core is also in idle.	R/W	0
5:3	RESERVED	See note.	R/W	000
2	IDLPER_ARM	Selects idle entry mode for external peripheral clock. (ARMPER_CK) 0: The peripheral clock remains active when the MPU core and TC enter the idle mode. 1: The peripheral clock is stopped in conjunction with the MPU core and TC clocks when the idle mode is entered. ARMPER_CK is no longer dependent only on ARM_IDLE conditions. As long as TC or MPU core is active, ARMPER_CK clock is on. When both are inactive, ARMPER_CK is shut off based on ARMPER IDLE/ACK.	R/W	0
1	IDLXORP_ARM	Selects idle entry mode for external reference peripheral clock ARMXOR_CK 0: The external peripheral clock ARMXOR_CK remains active when the MPU core enters the idle mode. 1: The external peripheral clock is stopped in conjunction with the MPU core clock when the idle mode is entered.	R/W	0
0	IDLWDT_ARM	Selects the idle entry mode for internal timer/watchdog connected to MPU core TIPB. When the timer/watchdog is configured as watchdog timer, the clock is never shutdown regardless of the IDLWDT_ARM bit. 0: The clock supplied to the timer/watchdog remains active when the MPU core enters idle mode. 1: The timer/watchdog clock is stopped in conjunction with the MPU core clock when the idle mode is entered.	R/W	0

Note: For reserved bits, reading gives undefined values. Writing to has no effect.

Table 72. MPU Core Idle Enable Control Register 2 (ARM_IDLECT2)

Base Address = 0xFFFE CE00, Offset = 0x08				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:12	RESERVED	See note.	R/W	0000
11	EN_CKOUT_ARM	This read-write bit enables the free running clock from DPLL1 output 0: The clock generated from DPLL1 output is stopped. This bit must be set to logic 1 to resume clock activity. 1: The clock generated from DPLL1 output is active.	R/W	0
10:9	RESERVED	See note.	R/W	00
8	DMACK_REQ	Forces the permanently-supplied-clock to the system DMA controller to function on a clock request basis 0: The DMA clock is shut down when the idle mode is entered, if IDLIF_ARM bit of ARM_IDLECTL1 is set. 1: The DMA clock is stopped by default and is reactivated upon DMA request only.	R/W	1
7	EN_TIMCK	Enables the MPU core internal timer clock connected to the MPU core TIPB 0: The MPU core timer clock is stopped. 1: The MPU core timer clock is active and can be stopped depending on the IDLTIM_ARM bit of ARM_IDLECTL1.	R/W	0
6	EN_APICK	Enables the clock of the MPUI 0: The MPUI clock is stopped. This bit must be set to logic 1 to enable clock activity. 1: The MPUI clock is active. The clock ON/OFF is now controlled as per IDLAPI_ARM bit.	R/W	0
5:4	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	00
3	EN_LCDCK	Enables the LCD controller clock connected to the MPU core TIPB 0: The LCD clock is stopped. 1: The LCD clock is active.	R/W	0

Note: For reserved bits, reading gives undefined values. Writing to has no effect.

Table 72. MPU Core Idle Enable Control Register 2 (ARM_IDLECT2) (Continued)

Base Address = 0xFFFE CE00, Offset = 0x08				
Bit	**Name**	**Function**	**R/W**	**Reset**
2	EN_PERCK	Enables the external peripheral clock 0: The external peripheral clock ARMPER_CK is stopped. 1: The external peripheral clock ARMPER_CK is active and can be stopped, depending on the IDLLPER_ARM bit.	R/W	1
1	EN_XORPCK	Enables the clock of the OS timer connected to MPU core TIPB and the external reference peripheral clock 0: The OS timer clock and the external peripheral clock are stopped. 1: The OS timer clock and the external peripheral clock are active and can be stopped, depending on the IDLXORP_ARM bit of ARM_IDLECTL1.	R/W	0
0	EN_WDTCK	Enables the clock of the timer/watchdog connected to MPU core TIPB (When the timer/watchdog is configured as a watchdog timer, the clock is never shut down regardless of the value of IDLWDT_ARM and EN_WDTCK.) 0: The timer/watchdog clock is stopped. 1: The clock supplied to timer/watchdog clock is active and can be stopped, depending on the IDLWDT_ARM bit of ARM_IDLECTL1.	R/W	0

Note: For reserved bits, reading gives undefined values. Writing to has no effect.

Table 73. MPU Core Restore Power Delay Register (ARM_EWUPCT)

Base Address = 0xFFFE CE00, Offset = 0x0C

Bit	Name	Function	R/W	Reset
31:6	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x000
5	REPWR_EN	Enables the external power control feature. 0: The $\overline{\text{FLASH.RP}}$ pin is set to logic low when TC is in idle mode. 1: The $\overline{\text{FLASH.RP}}$ pin stays high when the TC idle mode is entered.	R/W	1
4:0	EXTPWR	Defines the delay from $\overline{\text{FLASH.RP}}$ pin going high to the clocks restarting. Reference clock is the EMIFS CK_REF.	R/W	11111

Table 74. Master Software Reset Register (ARM_RSTCT1)

Base Address = 0xFFFE CE00, Offset = 0x10

Bit	Name	Function	R/W	Reset
31:4	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x000
3	SW_RST	Global system reset. Resets both the DSP core and the MPU core and peripherals. This bit is always read 0. 0: The DSP core, the MPU core, and the peripheral clock domains are enabled. 1: Resets the OMAP 3.2 hardware engine. Once set to logic 1 by the MPU core processor, this bit returns to logic 0 on the next cycle.	R/W	0
2	DSP_RST	Resets the priority registers (TIPB module), the EMIF configuration registers, and the MPUI control logic in the DSP core. This bit is set by the external reset pins and is released by writing a logic 1. 0: The priority registers, the EMIF configuration registers, and the MPUI are reset. 1: The priority registers and the EMIF configuration registers can be programmed.	R/W	0

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

Table 74. Master Software Reset Register (ARM_RSTCT1) (Continued)

Base Address = 0xFFFE CE00, Offset = 0x10				
Bit	**Name**	**Function**	**R/W**	**Reset**
1	DSP_EN	Resets the DSP core 0: Resets the DSP core, excluding the configuration setting. The reset state is maintained as long as this bit is asserted low. 1: The DSP core is enabled. After a global reset, this bit must be set to 1 in order to enable the DSP core megacell.	R/W	0
0	ARM_RST	Resets the MPU core. This bit is always read as 0. 0: The MPU core clock domain is enabled. 1: Reset the MPU core. Once set to 1 by the MPU core, this bit returns to 0 on the next cycle.	R/W	0

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

Table 75. Peripherals Reset Register (ARM_RSTCT2)

Base Address = 0xFFFE CE00, Offset = 0x14				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:1	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x0000
0	PER_EN	MPU core Peripheral reset. Resets and/or enables the external peripherals connected to MPU core TIPB (controls 3.2 ARMPER_nRST). 0: Resets MPU core peripherals. 1: Enables MPU core peripherals.	R/W	0

Table 76. MPU Core Clock Reset Status Register (ARM_SYSST)

Base Address = 0xFFFE CE00, Offset = 0x18				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:14	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	00
13:11	CLOCK_SELECT	Reading these bits indicates the clock_select pins and the current clocking mode selection. Writing to these bits enables a switch in the OMAP3.2 clocking scheme. These bits are at logic 0 after reset: 000: Fully synchronous 001: Reserved 010: Synchronous scalable 011: Reserved 100: Reserved 101: Bypass 110: Mix mode #3, MPU core synchronous to TC, DSP core MMU synchronous scalar to MPU core and TC 111: Mix mode #4, DSP core MMU synchronous to TC, MPU core synchronous scalar to DSP core MMU and TC	R/W	000
10:7	RESERVED	These read only bits are undefined.	R	0
6	IDLE_DSP	Indicates the DSP core state. 0: The DSP core is active. 1: The DSP core is in global-idle state.	R	0
5	POR	Indicates (in conjunction with 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. 0: No power-on-reset has been detected. 1: A power-on-reset has occurred.	R/C	1
4	EXT_RST	Indicates that 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. 0: No external reset has been detected. 1: An external reset has occurred.	R/C	1

Table 76. MPU Core Clock Reset Status Register (ARM_SYSST) (Continued)

Base Address = 0xFFFE CE00, Offset = 0x18				
Bit	**Name**	**Function**	**R/W**	**Reset**
3	ARM_MCRST	Indicates whether or not an MPU core reset has occurred. This bit is cleared to 0 upon an external reset pulse asserting at the CHIP_nRESET pin, or by writing to it a logic 0. This bit cannot be written to logic 1 from the TIPB interface. 0: The MPU core processor has not been reset. 1: The MPU core processor has been reset.	R/C	1
2	ARM_WDRST	Indicates whether or not the reset has been asserted due to an MPU core timer/watchdog underflow. This bit is cleared to 0 upon an external reset pulse asserting at the CHIP_$\overline{\text{RESET}}$ pin, or by writing to it a logic 0. This bit cannot be written to logic 1 from the TIPB interface. 0: An MPU core timer/watchdog underflow has not occurred. 1: An MPU core timer/watchdog underflow has generated the reset.	R/C	0
1	GLOB_SWRST	Indicates whether or not the reset has been asserted due to global software reset (DSP_EN set to 0 and ARM_RST set to 1). This bit is cleared to 0 upon an external reset pulse asserting at the CHIP_nRESET pin, or by writing to it a logic 0. This bit cannot be written to logic 1 from the TIPB interface. 0: Global software reset has not been requested. 1: Global software reset has been requested.	R/C	0
0	DSP_WDRST	Indicates whether or not the reset has been asserted due to DSP core timer/watchdog underflow. This bit cannot be written to logic 1 from the TIPB interface. 0: A DSP core timer/watchdog underflow has not occurred. 1: A DSP core timer/watchdog underflow has generated the reset.	R/C	0

Table 77. MPU core Clock Out Definition Register (ARM_CKOUT1)

Base Address = 0xFFFE CE00, Offset = 0x1C				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:6	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x0000
5:4	TCLKOUT	The POCLKOUT3 pin functions are: 00: Reserved 01: POCLKOUT3 pin is an output and reflects CK_GEN3 clocking signal 10: POCLKOUT3 pin is an output and reflects the TC_CK clock 11: Reserved POCLKOUT3 is not a target clock; it is a free-running clock to select CK_GEN3 or TC_CK frequencies.	R/W	01
3:2	DCLKOUT	The POCLKOUT2 pin functions are: 00: POCLKOUT2 pin is an output and reflects the DSPMMU_CK clock 01: POCLKOUT2 pin is an output and reflects the CK_GEN2 clocking signal 10: POCLKOUT2 pin is an output and reflects the DSP_CK clock 11: POCLKOUT2 pin is an output and reflects the low-frequency clock that supplies the internal watchdog timers (CK_REF/14) POCLKOUT2 is not a target clock; it is a free-running clock to select DSPMMU_CK, CK_GEN2, or DSP_CK frequencies.	R/W	01
1:0	ACLKOUT	The POCLKOUT1 pin functions are: 00: Reserved 01: POCLKOUT1 output pin reflects CK_GEN1 10: POCLKOUT1 output pin reflects ARM_CK 11: POCLKOUT1 output pin reflects the low-frequency clock that supplies the internal timers (CK_REF/14) POCLKOUT1 is not a target clock; it is a free-running clock to select CK_GEN1, ARM_CK, or CK_REF/14 clock frequency.	R/W	01

Table 78. MPU core Reserved Register (ARM_CKOUT2)

Base Address = 0xFFFE CE00, Offset = 0x20				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:0	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x0000

Table 79. MPU core Idle Enable Control Register 3 (ARM_IDLECT3)

Base Address = 0xFFFE CE00, Offset = 0x24				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:6	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x000
5	IDLTC2_ARM	Selects the idle entry mode for the TC2 clock 0: The TC2 clock remains active when the MPU core enters the idle mode (ARM_CK stopped). 1: The TC2 clock is stopped in conjunction with the MPU core clock when the idle mode is entered. Cutting off is based on IDLE/ACK protocol between CLKRST and peripherals outside OMAP.	R/W	0
4	EN_TC2_CK	Enables the TC2 clock. This is a generic clock supplied to peripherals outside of the OMAP boundary and is at the same frequency as the TC clock. 0: The TC2_CK clock is stopped. Ensure that all peripherals connected to TC2_CK are inactive before setting 0 on EN_TC2_CK. 1: The TC2_CK clock is active.	R/W	1
3	IDLTC1_ARM	Selects the idle entry mode for the TC1 clock 0: The TC1 clock remains active when the MPU core enters the idle mode (ARM_CK stopped). 1: The TC1 clock is stopped in conjunction with the MPU core clock when the idle mode is entered. Cutting off is based on IDLE/ACK protocol between CLKRST and peripherals outside OMAP.	R/W	0

Table 79. MPU core Idle Enable Control Register 3 (ARM_IDLECT3) (Continued)

Base Address = 0xFFFE CE00, Offset = 0x24				
Bit	**Name**	**Function**	**R/W**	**Reset**
2	EN_TC1_CK	Enables the TC1 clock. This is a generic clock supplied to peripherals outside of the OMAP boundary and is at the same frequency as the TC clock. 0: The TC1_CK clock is stopped. Ensure that all peripherals connected to TC1_CK are inactive before setting 0 on EN_TC1_CK. 1: The TC1_CK clock is active.	R/W	1
1	IDLOCPI_ARM	Selects the idle entry mode for the L3 OCP initiator 0: The L3 OCP-I clock remains active when the MPU core enters the idle mode. 1: The L3 OCP-I clock is stopped in conjunction with the MPU core clock when the idle mode is entered.	R/W	0
0	EN_OCPI_CK	Enables the L3 OCPI clock 0: The L3 OCPI clock is stopped. 1: The L3 OCPI clock is active.	R/W	1

4.4.2 DSP Core Registers

These registers are accessible by the DSP core or the MPU core. The DSP core control registers are 16-bit accessed. The offsets are given for byte addressing.

Table 80 lists the DSP core registers. Table 81 through Table 89 provide register bit descriptions.

Table 80. DSP Core Registers

Base Address = 0xE100 8000 or 0x008000			
Name	**Description**	**R/W**	**Offset**
DSP_CKCTL	DSP core clock control prescaler selection	R/W	0x00
DSP_IDLECT1	DSP core idle enable control 1	R/W	0x04
DSP_IDLECT2	DSP core idle enable control 2	R/W	0x08
DSP_EWUPCT	DSP core reserved register 1	R/W	0x0C
DSP_RSTCT1	DSP core reserved register 1	R/W	0x10
DSP_RSTCT2	DSP core peripherals reset	R/W	0x14

Table 80. DSP Core Registers (Continued)

Base Address = 0xE100 8000 or 0x008000			
Name	**Description**	**R/W**	**Offset**
DSP_SYSST	DSP core clock reset status	R/W	0x18
DSP_CKOUT1	DSP core reserved register 3	R/W	0x1C
DSP_CKOUT2	DSP core reserved register 4	R/W	0x20

Table 81. DSP Core Clock Control Prescaler Selection Register (DSP_CKCTL)

Base Address = 0xE100 8000 or 0x00 8000, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:9	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x00
8	TIMXO	Selects either a CK_GEN2 frequency clock or the input reference clock (CLK_REFIN) to supply timers 0: The DSPTIM_CK clock frequency is the input reference clock. 1: The DSPTIM_CK clock frequency is issued from CK_GEN2 divided by 2.	R/W	1
7	RESERVED	This bit must be set to 1.	R/W	1
6:5	RESERVED	These bits must be set to 00.	R/W	00
4	RESERVED	This bit must be set to 1.	R/W	1
3:2	RESERVED	These bits must be set to 00.	R/W	00
1:0	DSP_PERDIV	Defines the prescaler value from CK_GEN2 to the DSP core external peripheral clock 00: CK_GEN2 01: CK_GEN2/2 10: CK_GEN2/4 11: CK_GEN2/8	R/W	00

Table 82. DSP Core Idle Enable Control Register 1 (DSP_IDLECT1)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x04				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:9	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x00
8	IDLTIM_DSP	Selects the idle entry mode for the internal DSP core timer clock. 0: The DSPTIM_CK clock remains active when DSP core enters the idle mode. 1: The DSPTIM_CK clock is stopped in conjunction with DSP core clock when the idle mode is set.	R/W	0
7	RESERVED	This bit must be set to 0.	R/W	0
6	WKUP_MODE	This bit has no effect in the OMAP 3.2 hardware engine.	R/W	1
5	IDLDPLL_DSP	This bit has no effect in the OMAP 3.2 hardware engine.	R/W	0
4	IDLIF_DSP	This bit has no effect in the OMAP 3.2 hardware engine	R/W	0
3	RESERVED	This bit must be set to 0.	R/W	0
2	IDLPER_DSP	Selects the idle entry mode for the external peripheral clock. 0: The DSPPER_CK clock remains active when DSP core enters the idle mode. 1: The DSPPER_CK is stopped in conjunction with the DSP core clock when the idle mode is set.	R/W	0

Table 82. DSP Core Idle Enable Control Register 1 (DSP_IDLECT1) (Continued)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x04				
Bit	**Name**	**Function**	**R/W**	**Reset**
1	IDLXORP_DSP	Selects the idle entry mode for the external reference peripheral clock. 0: The DSPXOR_CK clock remains active when DSP core enter the idle mode. 1: The DSPXOR_CK clock is stopped in conjunction with the DSP core clock when the idle mode is set.	R/W	0
0	IDLWDT_DSP	Selects the idle entry mode for the internal timer/watchdog connected to DSP core TIPB. 0: The clock supplied to the timer/watchdog remains active when the DSP core enters the idle mode. 1: The timer/watchdog clock is stopped in conjunction with the DSP core clock when the idle mode is set. When the timer/watchdog is configured as a watchdog timer, the clock is never shut down regardless of the value of the IDLWDT_DSP bit.	R/W	0

Table 83. DSP Core Idle Enable Control Register 2 (DSP_IDLECT2)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x08				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:6	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x00
5	EN_TIMCK	Enables the internal DSP core timer clock (DSPTIM_CK). 0: DSPTIM_CK clock is stopped. 1: DSPTIM_CK clock is active and can be stopped depending on the IDLTIM_DSP bit of DSP_IDLECT1.	R/W	0
4	RESERVED	This bit must be set to 0.	R/W	0
3	RESERVED	This bit must be set to 0.	R/W	0
2	EN_PERCK	Enables external peripheral clock (DSPPER_CK). 0: DSPPER_CK clock is stopped. 1: DSPPER_CK clock is active and can be stopped, depending on the IDLPER_DSP bit of DSP_IDLECT1.	R/W	0

Table 83. DSP Core Idle Enable Control Register 2 (DSP_IDLECT2) (Continued)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x08				
Bit	**Name**	**Function**	**R/W**	**Reset**
1	EN_XORPCK	Enables the external reference clock (DSPXOR_CK). 0: DSPXOR_CK clock is stopped. 1: DSPXOR_CK clock is active and can be stopped, depending on the IDLXORP_DSP bit of DSP_IDLECT1.	R/W	0
0	EN_WDTCK	Enables the internal timer/watchdog clock (DSPWDG_CK). 0: DSPWDG_CK clock is stopped. 1: DSPWDG_CK clock is active and can be stopped, depending on the IDLWDT_DSP bit of DSP_IDLECT1.	R/W	0

Table 84. DSP Core Reserved Register 1 (DSP_EWUPCT)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x0C				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x0

Table 85. DSP Core Reserved Register 2 (DSP_RSTCT1)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x10				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x0

Table 86. DSP Core Peripherals Reset Register (DSP_RSTCT2)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x14				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:2	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x0000

Table 86. DSP Core Peripherals Reset Register (DSP_RSTCT2) (Continued)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x14				
Bit	**Name**	**Function**	**R/W**	**Reset**
1	WD_PER_EN	Controls the WD_DSPPER_nRST output, which can be used to reset the external peripherals connected to DSP core. This WD_DSPPER_nRST pin is also reset by an event on the DSP core watchdog timer. 0: Sets the WD_DSPPER_nRST pin to a low-level output voltage. 1: Sets the WD_DSPPER_nRST pin to a high-level output voltage.	R/W	0
0	PER_EN	Controls the DSPPER_nRST output, which can be used to reset the external peripherals connected to DSP core TIPB. 0: Sets the DSPPER_nRST pin to a low-level output voltage. 1: Sets the DSPPER_nRST pin to a high-level output voltage.	R/W	0

Table 87. DSP Core Clock Reset Status Register (DSP_SYSST)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x18				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:14	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	00
13:11	CLOCK_SELECT	These read-only bits reflect the CLOCK_SELECT pins and indicate the current clocking mode selection. 000: Fully synchronous 001: Reserved 010: Synchronous scalable 011: Reserved 100: Reserved 101: Bypass 110: Mix mode #3, MPU core synchronous to TC, DSP core MMU synchronous scalar to MPU core and TC 111: Mix mode #4, DSP core MMU synchronous to TC, MPU core synchronous scalar to DSP core MMU and TC	R	000

Table 87. DSP Core Clock Reset Status Register (DSP_SYSST) (Continued)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x18				
Bit	**Name**	**Function**	**R/W**	**Reset**
10:7	RESERVED	Reserved bits	R	0000
6	IDLE_ARM	Indicates the MPU core state 0: The MPU core is active. 1: The MPU core is in idle state.	R	0
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. 0: No power-on reset has been detected. 1: A power-on reset has occurred.	R/C	1
4	EXT_RST	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. 0: No external reset detected. 1: An external reset has been asserted.	R/C	1
3	DSP_ARM_RST	Used by the DSP core to hold the MPU core in reset. This is for test and debug purposes only. Not for users. 0: The MPU core is enabled. 1: Reset the MPU core.	R/W	0
2	ARM_WDRST	Indicates whether or not the reset has been asserted due to a MPU core timer/watchdog underflow. This bit cannot be written to logic 1 from the TIPB interface. 0: An MPU core timer/watchdog underflow has not occurred. 1: An MPU core timer/watchdog underflow has generated the reset.	R/C	0

Table 87. DSP Core Clock Reset Status Register (DSP_SYSST) (Continued)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x18				
Bit	**Name**	**Function**	**R/W**	**Reset**
1	GLOB_SWRST	Indicates whether or not the reset has been asserted due to a global software reset. This bit cannot be written to logic 1 from the TIPB interface. 0: A global software reset has not been requested. 1: A global software reset has been requested.	R/C	0
0	DSP_WDRST	Indicates whether or not the reset has been asserted due to a DSP core timer/watchdog underflow. This bit cannot be written to logic 1 from the TIPB interface. 0: A DSP core timer/watchdog underflow has not occurred. 1: A DSP core timer/watchdog underflow has generated the reset.	R/C	0

Table 88. DSP Core Reserved Register 3 (DSP_CKOUT1)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x1C				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x0000

Table 89. DSP Core Reserved Register 4 (DSP_CKOUT2)

Base Address = 0xE100 8000 or 0x008000, Offset = 0x20				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	RESERVED	Reading these bits gives undefined values. Writing to them has no effect.	R/W	0x0000

4.4.3 DPLL Registers

Table 90 lists the 16-bit DPLL registers, Table 91 and Table 92 describe the register bits.

Table 90. DPLL Registers

Base Address = 0xFFFE CF00			
Name	**Description**	**R/W**	**Offset**
DPLL1_CTL_REG	DPLL1 control	R/W	0x00
DPLL2_CTL_REG	DPLL2 control (all reserved)	R	0x100

Table 91. DPLL1 Control Register (DPLL1_CTL_REG)

Base Address = 0xFFFE CF00, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
15	LS_DISABLE	Controls the level shifter power-down pin. 0: Level shifter is in *transparent* mode; all signals between the wrapper and the DPLL core are connected. 1: Level shifter is in *isolated* mode; the wrapper and the DPLL core are disconnected, so the DPLL core power supply (VDD_DPLL) can be turned off. There is no leakage current between VDD and VDD_DPLL.	R/W	0
14	IAI	Initialize after idle. Value of this bit must not be changed. Must be set to 0.	R/W	0
13	IOB	Initialize on break. When high, DPLL switches to bypass mode and starts a new locking sequence, even if the DPLL core indicates that it has lost the lock. When low, DPLL continues to output the synthesized clock, even if the core indicates it has lost the lock but the BREAKLN is active low.	R/W	1
12	TEST†	Controls the test output clock on the DPLL_TCLKOUT pin as given below: 0: DPLL_TCLKOUT = DPLL1 output clock when in test mode. 1: DPLL_TCLKOUT = DPLL1 output clock divided by 32 when in test mode. X: DPLL_TCLKOUT = 0 when not in test mode.	R/W	0
11:7	PLL_MULT	DPLL multiply value. The maximum clock out frequency is 31 * CK_REF.	R/W	00000
6:5	PLL_DIV	DPLL divide value. The minimum DPLL1 clock out frequency is CK_REF/4. 00: CLKOUTDPLL1 output clock = CK_REF 01: DPLL1 output clock CLKOUT = CK_REF/2 10: DPLL1 output clock CLKOUT = CK_REF/3 11: DPLL1 output clock CLKOUT = CK_REF/4	R/W	00

† POCLKOUT[1, 2, 3] is a nongated output from clock domain[1, 2, 3]. Using the [A/D/T] CLKOUT bits, select between different clocks used in the respective domains.

Table 91. DPLL1 Control Register (DPLL1_CTL_REG) (Continued)

Base Address = 0xFFFE CF00, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
4	PLL_ENABLE	Requests that the DPLL enter the lock mode. DPLL enters the lock mode only after it has synthesized the desired frequency. 0: DPLL enters the bypass mode. 1: DPLL enters the lock mode.	R/W	0
3:2	BYPASS_DIV	Determines the clock out frequency when in bypass mode. 00: DPLL1 output clock CLKOUT = CK_REF 01: DPLL1 output clock CLKOUT = CK_REF/2 1X: DPLL1 output clock CLKOUT = CK_REF/4	R/W	00
1	BREAKLN	Indicates whether DPLL has broken lock for some unknown reason. 0: DPLL has broken lock for some unknown reason. 1: Lock condition is restored or a write to a control register occurs.	R	0
0	LOCK	Indicates if DPLL is in lock mode and the clock out has the desired synthesized frequency. 0: DPLL is in bypass mode. 1: DPLL is in lock mode.	R	0

† POCLKOUT[1, 2, 3] is a nongated output from clock domain[1, 2, 3]. Using the [A/D/T] CLKOUT bits, select between different clocks used in the respective domains.

Table 92. DPLL2 Control Register (DPLL2_CTL_REG)

Base Address = 0xFFFE D000, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset Value**
15:0	RESERVED	Reserved. Do not write to these bits.	R	0x00002000

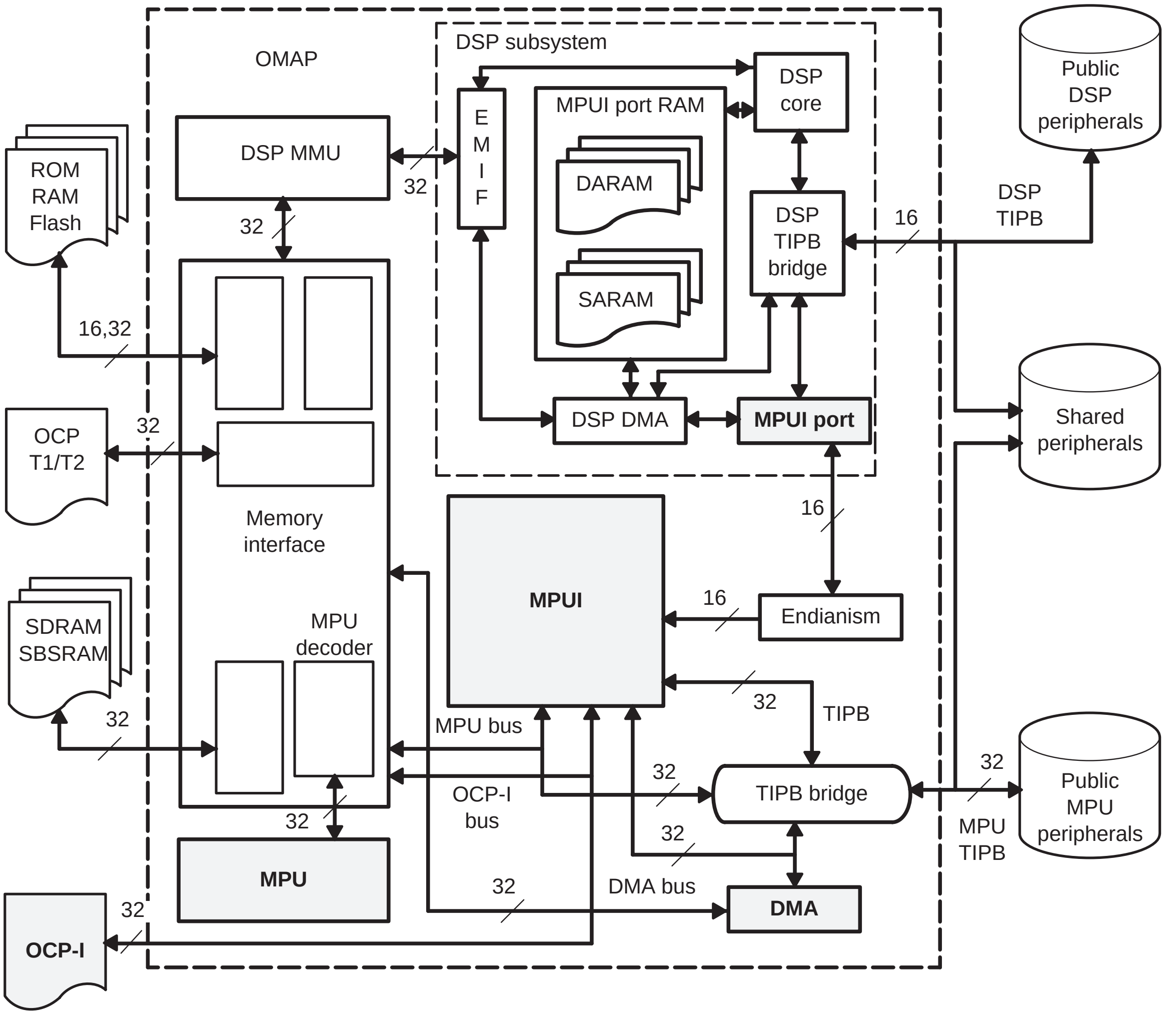
5 MPU Core and MPUI Port

The MPU core, system DMA, and OCP initiator (OCP-I) can access the DSP core memories and peripherals via two interfaces: the MPUI and the MPUI port. The MPUI and the MPUI port have distinct features and functions:

- ❑ The MPUI is a module in the MPU core subsystem that connects to the MPUI port.
- ❑ The MPUI port is a module contained in the DSP core subsystem.

Figure 8-1 shows the MPU core- and DSP core-relevant modules and connections. In this figure, DMA represents the system DMA.

Figure 42. OMAP 3.2 MPUI and MPUI Port Environment



5.1 MPUI

The MPUI module connects the MPU core, system DMA, and OCP-I to the MPUI port. The MPUI allows sharing of the DSP core internal memories and peripherals with the MPU core/system DMA/OCP-I.

The MPUI module supports two access modes: host-only mode (HOM) and shared access mode (SAM).

In HOM, the MPUI can access the SARAM and DSP core peripherals. An MPUI port RAM configuration register can set the range of the SARAM (host-only RAM) to which the MPUI has exclusive access. The DSP core is denied access to the host-only RAM portion; however, both the MPUI and the DSP core can access the other part of the SARAM (shared-access RAM). All accesses available to the MPUI in HOM remain available, even if the DSP core is in idle mode. The MPUI does not have access to the DARAM in HOM. The DSP core TIPB peripherals are accessible only by MPUI in HOM.

In SAM, the entire MPUI port RAM and DSP core peripherals are accessible by the MPUI and DSP core. If both the DSP core and the MPUI are accessing the same memory or peripheral at the same time, priority is given to the DSP core. The access is synchronized to the internal DSP core clock. Note that SARAM and the DSP core TIPB peripherals can be accessed by the MPUI both in HOM and SAM. The DARAM and the EMIF, however, can be accessed by the MPUI only in SAM.

HOM is more efficient than the SAM because no synchronization is involved. However, the HOM depends on the host operating frequency, which is normally slower than the internal DSP core clock. The system software can switch between HOM and SAM, or vice versa, if desired, and it is up to the software to manage the system resources.

Sections 5.1.1 through 5.1.7 describe the MPUI functions.

5.1.1 Access Request

If the MPU core, OCP-I, and system DMA request access to the DSP core memory/peripherals at the same time, the MPUI gives priority to one of the three, based on the ACCESS_PRIORITY bits of (MPUI_CONTROL). The programmable priority scheme must be configured during the system boot process. When the MPUI initiates an MPUI port access, it must wait until the access is completed before starting a new one. Pipelining is not supported by the MPUI.

The MPUI supports 8-, 16-, or 32-bit access requests, even though the interface from the MPUI to the MPUI port is a 16-bit interface. The MPUI, on receiving the 32-bit access request from the MPU core, OCP-I, or system DMA, initiates two 16-bit MPUI accesses, one after the other, one with address X and other with address X + 2. On a 32-bit read access, The MPUI packs the data from the two transactions and sends a signal to the MPU core/OCP-I/system DMA once the second 16-bit read is completed.

5.1.2 Endian Conversion

Because the MPUI uses little endian ordering and the MPUI port uses big endian ordering, there is an endianism conversion block between the MPUI and the MPUI port. This endianism block converts data between the little and big endian formats. Software controls the swapping logic, giving it maximum flexibility to handle different types of data, based on the BYTE_SWAP_CTL and WORD_SWAP_CTL bits of (MPUI_CONTROL) from the MPUI.

Endianism conversion capability makes peripheral (control) register read/writes more convenient. The DSP core memory read/write transactions require conversion of the data, but it is more convenient for the MPU core, OCP-I, and system DMA controller to read and write to registers without conversion.

The endianism conversion logic only performs byte (8-bit) or 2-byte (16-bit) swaps from big endian to little endian and vice versa without address manipulation. The swapping mode can be programmed differently according to information type, information source, and information direction, such as 32-bit or 16-bit data, DSP core access, or DSP DMA access and read or write.

Since the MPUI port does not allow accessing with 32-bit data, the MPUI port logic converts two 16-bit data accesses sequentially when the MPUI accesses it with 32-bit data. The 2-byte swap is performed during sequential access conversion.

Table 93 and Table 94 show examples of endian conversion.

Table 93. Data Swap for 32-Bit Access for MPUI Port

32-Bit Accesses	BYTE_SWAP_CTL	WORD_SWAP_CTL
INPUT DATA	AA BB CC DD	AA BB CC DD
OUTPUT DATA	DD CC BB AA	CC DD AA BB

Table 94. Data Swap for 16-Bit Access for MPUI Port

16-Bit Accesses	BYTE_SWAP_CTL	WORD_SWAP_CTL
INPUT DATA	00 11	00 11
OUTPUT DATA	11 00	00 11

5.1.3 MPUI Strobe and Access Factor

The MPUI output strobe is an internal signal used to enable bus transactions between the MPUI and the MPUI port. The strobe active (low) pulse defines the beginning and end of a transaction. Because the MPUI may be required to communicate through the MPUI port with peripherals of varying speeds, a means to adjust the MPUI output strobe timing is available in between the MPUI and the MPUI port. The strobe active (low) pulse defines the beginning and end of a transaction.

To allow slow peripherals to answer, it is possible to stretch an access over 2 * *n* MPUI clock cycles using the ACCESS_FACTOR bits in (MPUI_CONTROL). The peripheral then has *n* clock cycles to answer (n cycles the strobe is high; n cycles the strobe is low). Note that the MPUI clock referenced here is the input clock reference to the MPUI module generated from the clock and reset management module. The clock rate for the MPUI is fixed to the same value as that of the OMAP 3.2 traffic controller module. For more detail, see section 4.

5.1.4 MPUI Port RAM Access

In HOM, only the MPU core, system DMA, or OCP-I can access the SARAM through the MPUI. Although the entire SARAM is accessible, the MPU core must first set the accessible size of the SARAM. The API_SIZE bit field in the register (DSP_MPUI_CONFIG) sets the size using the formula, *(integer value of API_SIZE * 8K bytes*), starting from the first SARAM block. For details on available SARAM memory space and SARAM start addresses, see the memory map in the *OMAP5912 Data Manual* (SPRS231). For details on the translation of DSP core internal (logical) addresses into OMAP (physical) addresses, see *OMAP5912 Multimedia Processor DSP Subsystem Reference Guide* (SPRU750).

The host can not access the SARAM before releasing the MPU core reset. After releasing the MPU core reset and before releasing the DSP core reset, the DSP core is in HOM and all the SARAM is accessible only by the host as the default API_SIZE value is 0xFFFF. Then the (DSP_MPUI_CONFIG) can be programmed to give the host exclusive access to a portion or to all the SARAM. After the DSP core reset is released, the DSP core is automatically changed to SAM; consequently, whatever the value of the (DSP_MPUI_CONFIG) , all the SARAM is shared between the DSP core and the host. DSP_MPUI_CONFIG must be set when the ARM_RSTCT1.DSP_RST register bit is 0, and when MPU core reset is deasserted. If DSP_MPUI_CONFIG changes while the ARM_RSTCT1.DSP_RST bit is 1, then ARM_RSTCT1.DSP_RST must be cleared to 0 again before the new DSP_MPUI_CONFIG value can take effect.

In HOM, the SARAM memory requests are completely asynchronous relative to the DSP core clock. Therefore, memory accesses can be performed without resynchronization, allowing faster communication between MPU core/system DMA/OCP-I and SARAM. Any access to DARAM or EMIF causes a bus error.

In SAM, both DSP core and MPU core/system DMA/OCP-I can access the entire SARAM, DARAM, and EMIF. If API_SIZE in the DSP_MPUI_CONFIG register equals 0xFFFF, the DSP core can not access the SARAM. The DARAM is located at the byte address range from 0x000000 to 0x00FFFF in DSP core memory space. In this mode, the asynchronous host accesses from the MPUI are resynchronized on the DSP core clock internally in the MPUI port logic. If MPUI and DSP core attempt to access the same memory block, it causes a conflict between MPUI and DSP core accesses. The DSP core has the priority and the MPUI access is not acknowledged and is delayed by one or several cycles. Nevertheless, if an MPUI cycle begins before a DSP core request, the cycle is finished before recognizing the DSP core.

5.1.5 Peripheral Access

In HOM, only the MPU core/system DMA/OCP-I can access the DSP core shared peripherals. To determine the addresses of the peripherals. Peripheral requests are completely asynchronous relative to DSP core clocks. Therefore, peripheral accesses can be performed without any wait states, allowing faster communication between the MPU core and peripherals. In the HOM, the MPUI port is a simple bridge between the MPU core and DSP core TIPB bridges for the address, data, and control signals. The ACCESS_FACTOR bits of (MPUI_CONTROL) control the access clock that synchronizes the transfer between the MPUI and the MPUI port.

In SAM, both the DSP core and the MPU core/system DMA/OCP-I can access the DSP core peripherals. However, MPU core/system DMA/OCP-I can only access the DSP core shared peripherals, and any MPUI access to DSP core private peripherals causes a time-out error.

In SAM, the asynchronous host accesses from the MPU core/system DMA/OCP-I are resynchronized internally in the DSP core logic. If MPUI and DSP core attempt to access the same peripheral, it causes a conflict between MPUI and DSP core accesses. The DSP core has the priority and the MPUI waits one or several cycles. Nevertheless, if a MPUI transfer begins before a DSP core request, the access is completed before recognizing the DSP core.

5.1.6 MPUI and DSP Core TIPB Bridge Time-Out

When operating in HOM, an MPUI access time-out limits the maximum time a peripheral can stall the processor. When starting an access on the DSP core TIPB, the time-out counter is loaded with the value programmed in the TIMEOUT bits of (MPUI_CONTROL). If the current cycle is not finished when the counter reaches 0, the MPUI terminates the access and an abort exception is generated to the MPU core/system DMA/OCP-I. The maximum value for time-out is 256 MPUI clock cycles. Therefore, if the MPUI clock is 50 MHz (equal to the traffic controller clock frequency setting), maximum timeout is 256 x 20 ns, or 5.12 μs. A value of 0 can disable the time-out for debug and test purposes.

In SAM, MPUI time-out is automatically disabled to avoid a time-out conflict with DSP core TIPB bridge time-out. In SAM, the DSP core TIPB bridge can time-out a peripheral access if a predetermined period has elapsed and no response has been received from the peripheral. The DSP core TIPB bridge generates a time-out error to MPUI through the MPUI port.

5.1.7 Debug

Three registers are provided for debug capability: (DEBUG_ADDRESS), (DEBUG_DATA), and (DEBUG_FLAG).

- (DEBUG_ADDRESS) indicates the location of a malfunction. If an access is aborted or has a size mismatch, the address of the access is saved in (DEBUG_ADDRESS).
- (DEBUG_DATA) stores the data of the malfunction. If a read access has a size mismatch, the read value is saved into (DEBUG_DATA). However, the value in this register is irrelevant when the read access is aborted. When a write access is aborted or has a size mismatch, the write value is saved in (DEBUG_DATA).
- The cause of a malfunction is always reported to (DEBUG_FLAG). This register also shows which host (MPU core/system DMA/or OCP-I) is responsible for the abort.

5.2 MPUI Port

The MPUI port is an interface between DSP core resources and the MPUI. The MPUI port accesses the MPUI port RAM via the DSP DMA, and it accesses the DSP core shared peripherals via the DSP core TIPB bridge.

Sections 5.2.1 through 5.2.8 describe the MPUI port functions.

5.2.1 Access Modes

The MPUI port offers HOM and SAM based on the DSP core resource. The modes are as follows:

- HOM_M (host only for SARAM access)
- HOM_R (host only for DSP core TIPB access)
- SAM_M (shared access for SARAM, DARAM, EMIF access)
- SAM_R (shared access for DSP core TIPB access)

Only the DSP core can select HOM or SAM. This is controlled by the SMOD bits of (APIRS). Note that the MPU core cannot write these bits, but it can read these bits.

Since SMOD has two bits, the following four mode combinations are available:

- SAM for DSP core memory and TIPB. The host and the DSP core can access the SARAM, DARAM, EMIF, and DSP core TIPB peripherals.
- SAM for DSP core memory and HOM for DSP core TIPB. Only by the host can access the DSP core TIPB peripherals via the DSP core TIPB bus. The DSP core is denied access to all the peripherals. SARAM, DARAM, and EMIF are accessible by the host and the DSP core.
- HOM for DSP core memory and SAM for DSP core TIPB. The host can configure the SARAM for its exclusive access. The DSP core can only access the nonexclusive part of SARAM. The DSP core TIPB peripherals are accessible by the host and the DSP core.
- HOM for DSP core memory and TIPB. The host can configure the SARAM for its exclusive access. The DSP core can only access the nonexclusive part of SARAM. Only the host can access the DSP core TIPB peripherals.

When the DSP core reset is activated, HOM_M and HOM_R are automatically selected. Thus, only the host can access the SARAM and DSP core TIPB peripherals at DSP core reset. This allows programs or data to be downloaded to SARAM, peripherals can be configured even during reset. When the DSP core reset is released, SAM_M and SAM_R are automatically selected. Thus, the host and the DSP core can access the SARAM, DARAM, EMIF, and DSP core TIPB peripherals.

5.2.2 Memory Accesses in HOM

In HOM_M mode, only the MPU core/system DMA/OCP-I can exclusively access the specified portion of SARAM with its range controlled by the API_SIZE bits of (DSP_MPUI_CONFIG). Memory requests are completely asynchronous relative to the DSP core clock. Therefore, memory accesses can be performed without any wait states, allowing faster communication between MPUI and SARAM. In HOM, the MPUI port is a simple bridge between MPU core/system DMA/OCP-I and SARAM memory bank for the address, data, and control signals.

5.2.3 Memory Accesses in SAM

When the MPUI port is in SAM_M mode, both the DSP core and the MPU core/system DMA/OCP-I can access the SARAM, DARAM, and EMIF. In this mode, the asynchronous host accesses from the MPU core/system DMA/OCP-I are resynchronized on the DSP core clock internally in the MPUI port logic. If the MPU core/system DMA/OCP-I and the DSP core attempt to access same memory block, it causes a conflict between the MPU core/system DMA/OCP-I and the DSP core accesses. The DSP core has the priority and the MPU core/system DMA/OCP-I access is not acknowledged and is delayed by one or more cycles.

However, if an MPU core/system DMA/OCP-I transfer has started before the DSP core request is received, the MPU core/system DMA/OCP-I transfer is completed before the DSP core access is initiated.

5.2.4 Peripheral Accesses in HOM

In HOM_R mode, only the MPU core/system DMA/OCP-I can access the DSP core shared peripherals and peripheral requests are completely asynchronous to DSP core clock. Therefore, peripheral accesses can be performed without any resynchronization, allowing for faster communication between the MPU core/system DMA/OCP-I and the peripherals. In this mode, the MPUI port is a simple bridge between the MPU core/system DMA/OCP-I and the DSP core TIPB bridge for the address, data, and control signals.

- Any DSP core access to the shared peripherals generates a bus error from the DSP core TIPB bridge.

- An MPUI access to a DSP core private peripheral causes a time-out error.

5.2.5 Peripheral Accesses in SAM

When the MPUI port is in SAM_R mode, both the DSP core and the MPU core/system DMA/OCP-I can access the peripherals. In this mode, the asynchronous host accesses from the MPU core/system DMA/OCP-I are resynchronized internally in the MPUI port logic. If the MPU core/system DMA/OCP-I and the DSP core attempt to access same peripheral at the same time, it causes a conflict between the MPU core/system DMA/OCP-I and the DSP core accesses. The DSP core has the priority and the MPU core/system DMA/OCP-I waits one or more cycles.

However, if an MPU core/system DMA/OCP-I transfer has begun before a DSP core request is received, the MPU core/system DMA/OCP-I transfer is completed before the DSP core access is initiated. The goal is to ensure a smooth transition, with no spurious reads or writes, and no lost reads or writes.

5.2.6 Posted Write Mode

System performances can be enhanced by enabling the posted write mode when the MPUI port RAM and DSP core TIPB are in SAM. The MPU core/system DMA/OCP-I write accesses are released before the write completion on the peripheral side. The MPU core/system DMA/OCP-I is then free to carry on with the next access and the posted writes run as slots become available. This functionality can be turned off to aid in debugging, because when write posting is enabled it becomes difficult to attribute a bus error to a particular access. Only the MPU core has full control over the posted write enable bits (ENA_WPOST_APIRAM and ENA_WPOST_TIPB in (APIRS)). At reset, posted write is inactive (0), but, when active, the posted write mode is selected for memory and peripheral accesses. Posted writes do not slow down the host; this is useful because DSP core has the priority in SAM. When the transition from SAM to HOM is issued, it occurs after any outstanding posted writes are completed.

5.2.7 Bus Error

There are two sources of bus error for the MPUI port: the MPUI port itself and the DSP core TIPB bridge. (DEBUG_FLAG) stores these bus errors.

- The MPUI port issues an abort to the MPU core upon detecting an incorrect bus transaction, such as two chip-selects being active at the same time, or if a write-only register, such as the interrupt register, is read in the MPUI port. The same is true if a read-only register is written.
- The DSP core TIPB bridge limits the time allowed for any peripheral bus transaction. The DSP core TIPB bridge can terminate a peripheral bus transaction if a predetermined period of time has elapsed and no response has been received from the peripheral. The DSP core TIPB bridge will issue an abort to the MPU core via the MPUI port. The DSP core TIPB bridge can also issue an abort to MPU core when the MPU core/system DMA/OCP-I addresses a peripheral in the wrong mode (performing an 8-bit access to a 16-bit peripheral or vice versa).

5.2.8 Interrupt

The DSP core can send one interrupt to the MPU core. This is done by setting the corresponding bit in (ST3, bit 12 of status register 3 of TMS320C55x DSP core− see TMS320C55x functional specification) to active-low.

Similarly, the MPU core can interrupt the DSP core, and eight DSP core interrupt request lines are mapped. An interrupt is generated when the MPU core writes a 0 to any of the MCU_IRQ[7:0] bits of (APIRI). These bits are written only by the MPU core or OCP-I.

5.3 MPUI Port and MPUI Registers

This section provides information about the MPUI port and MPUI registers. Table 95 lists the 16-bit MPUI port registers. Table 96 and Table 97 describe the register bits.

Table 95. MPUI Port Registers

Base Address = 0xE102 0000			
Name	**Description**	**R/W**	**Offset**
APIRI	MPUI port interrupt	R/W	0x00
APIRS	MPUI port control/status	R/W	0x02

Table 96. MPUI Port Interrupt Register (APIRI)

Base Address = 0xE102 0000, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:8	Reserved			
7:0	MPU_IRQ [7:0]	Interrupt flag register for interrupts from the MPU core/OCP-I to the DSP core. Active low. Only the host can set these bits. A DSP core interrupt is generated when the MPU core writes a zero to the MPU_IRQ bits. MPU_IRQ is automatically reset by MPUI port internal logic.	W by MPU core/system DMA/OCP-I No access by DSP core	0xFF

Table 97. MPUI Port Control/Status Register (APIRS)

Base Address = 0xE102 0000, Offset = 0x02				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:4	Reserved			
3	ENA_WPOST_ APIRAM	Enables posted write for writes to the MPUI port RAM. Available in SAMs only. 0: Posted write disabled. 1: Posted write enabled.	R/W by MPU core/system DMA/OCP-I No access by DSP core	0

Table 97. MPUI Port Control/Status Register (APIRS) (Continued)

Base Address = 0xE102 0000, Offset = 0x02				
Bit	**Name**	**Function**	**R/W**	**Reset**
2:1	SMOD [1:0]	HOM or SAM setting for MPUI port RAM and DSP core TIPB peripherals. 00: SAM for MPUI port RAM and DSP core TIPB. DSP core and MPU core/system DMA/OCP-I can access MPUI port RAM and DSP core TIPB. 01: HOM for DSP core TIPB: DSP core TIPB peripherals are accessible from MPU core/system DMA/OCP-I only, SAM for MPUI port RAM. 10: HOM for MPUI port RAM: MPUI port RAM is accessible from MPU core/system DMA/OCP-I only, SAM for DSP core TIPB peripherals. 11: HOM for MPUI port RAM and DSP core TIPB: MPUI port RAM and DSP core TIPB are accessible from MPU core only. A DSP core write to host only resources is not performed, and any DSP core read or write results in an abort.	R by MPU core/system DMA/OCP-I R/W by DSP core	11
0	ENA_WPOST_ TIPB	Enables posted write for writes to the DSP core TIPB peripherals. Available in SAMs only. 0: Posted write disabled. 1: Posted write enabled.	R/W by MPU core/system DMA/OCP-I No access by DSP core	0

5.3.1 MPUI Registers

All MPUI registers are 32-bit and are accessible only by MPU core/system DMA/OCP-I. Read access can be performed in the user mode. Table 98 lists the 32-bit MPUI registers. Table 99 through Table 108 provide register bit descriptions.

Table 98. MPUI Registers

Base Address = 0xFFFE C900			
Name	**Description**	**R/W**	**Offset**
MPUI_CONTROL	MPUI control register	R/W	0x00
DEBUG_ADDRESS	Debug address register	R/W	0x04
DEBUG_DATA	Debug data register	R/W	0x08
DEBUG_FLAG	Debug flag register	R/W	0x0C

Table 98. MPUI Registers (Continued)

Base Address = 0xFFFE C900			
Name	**Description**	**R/W**	**Offset**
MPUI_STATUS	MPUI status register	R/W	0x10
DSP_STATUS	DSP core status register	R/W	0x14
DSP_BOOT_CONFIG	DSP core boot configuration register	R/W	0x18
DSP_MPUI_CONFIG	MPUI port RAM configuration register	R/W	0x1C
DSP_MISC	DSP core miscellaneous	R/W	0x20
MPUI_ENHANCED_ CTRL	MPUI enhanced control register	R/W	0x24

Table 99. MPUI Control Register (MPUI_CONTROL)

Base Address = 0xFFFE C900, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:23	Reserved			
22:21	WORD_SWAP_CTL	Bits to control word (16-bit swap) between MPUI and MPUI port for a 32-bit access: 00: Word swap for all accesses (OMAP 3.0) 01: Word swap only for DSP core TIPB peripheral and MPUI port control/ status register accesses 10: Word swap only for MPUI port RAM accesses 11: Turn off word swap for all accesses	R/W	00
20:18	ACCESS_PRIORITY	MPUI access priority between MPU core, OCP-I, and system DMA requests: (Note: The lower the number, the higher the priority) 000: MPU → 1 DMA → 2 OCP-I → 3 001: MPU → 1 DMA → 3 OCP-I → 2 010: MPU → 2 DMA → 1 OCP-I → 3 011: MPU → 2 DMA → 3 OCP-I → 1 1X0: MPU → 3 DMA → 1 OCP-I → 2 1X1: MPU → 3 DMA → 2 OCP-I → 1	R/W	000

Note: For 32-bit words, enabling byte swaps with BYTE_SWAP_CTL does the following:
(1) Performs a 16-bit word swap
(2) Within those 16-bit words, a byte swap occurs

Table 99. MPUI Control Register (MPUI_CONTROL) (Continued)

Base Address = 0xFFFE C900, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
17:16	BYTE_SWAP_CTL	Bits to control byte swap between MPUI and MPUI port (see note below): 00: Turn off byte swap for all accesses 01: Byte swap only for DSP core TIPB peripheral and MPUI port control/ status register accesses 10: Byte swap for all accesses 11: Byte swap only for MPUI port RAM accesses	R/W	11
15:8	TIMEOUT	MPUI port peripherals access time-out. When starting a HOM access to a DSP core TIPB peripheral, the time-out counter is loaded with this value. If the current access is not finished when the counter reaches 0, the cycle is aborted and abort indications are given to the MPU core/OCP-I/system DMA. Maximum value for time-out is 255. Note: TIMEOUT setting in MPUI is applicable only for HOM peripheral accesses.	R/W	0xFF
7:4	ACCESS_FACTOR	Division factor of MPUI output strobe signal from MPUI. Allows access of slow peripherals by reducing the access frequency (extending active low pulse) of the internal signal used to enable bus transactions between MPUI and MPUI port. Access factor = 0: Equivalent to access factor = 1. Access factor ≠ 0: Number of MPUI clock periods that the MPUI output strobe remains active (low) waiting for MPUI port to return a ready-to-send-or- receive-data signal. Note: Access factor setting in MPUI is applicable only for HOM memory/peripheral accesses.	R/W	0x1

Note: For 32-bit words, enabling byte swaps with BYTE_SWAP_CTL does the following:
(1) Performs a 16-bit word swap
(2) Within those 16-bit words, a byte swap occurs

Table 99. MPUI Control Register (MPUI_CONTROL) (Continued)

Base Address = 0xFFFE C900, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
3	API_ERR_EN	Send MPUI port abort. 0: Mask the abort. 1: An abort signal is forwarded by MPUI to the MPU core for an aborted MPUI port transaction. API_ERR_EN only affects MPUI port aborts. It does not enable/disable aborts to MPU core/OCP-I/system DMA for aborts generated directly by MPUI. For instance, a timeout-generated abort in the MPUI is not affected by API_ERR_EN setting.	R/W	1
2	Reserved			1
1	TIMEOUT_EN	Enable or disable TIMEOUT functionality as characterized by register bits (15:8). 0: Do not enable TIMEOUT functionality. 1: Enable TIMEOUT functionality.	R/W	1
0	Reserved		R/W	1

Note: For 32-bit words, enabling byte swaps with BYTE_SWAP_CTL does the following:
(1) Performs a 16-bit word swap
(2) Within those 16-bit words, a byte swap occurs

Table 100. Debug Address Register (DEBUG_ADDRESS)

Base Address = 0xFFFE C900, Offset = 0x04				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:24	Reserved			
23:0	ADR_SAV	Bits 23-0 of address bus from MPU core, OCP-I, or system DMA interface are saved when abort or access size mismatch occurs.	R	0xFFFFFF

Table 101. Debug Data Register (DEBUG_DATA)

Base Address = 0xFFFE C900, Offset = 0x08				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:0	DATA_SAV	The value of MPU core/OCP-I/system DMA data input bus is saved when a read access has a size mismatch, and the MPU core data output bus is saved when a write access is aborted or has a size mismatch. If a read access is aborted, the value of this register is irrelevant.	R	0xFFFFFFFF

Table 102. Debug Flag Register (DEBUG_FLAG)

Base Address = 0xFFFE C900, Offset = 0x0C				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:15	Reserved		R/W	0x0000
14:13	HOST_ID	Indicates the host responsible for the abort: 00: MPU core 01: System DMA 10: OCP-I 11: Reserved	R	00
12:11	SMOD_SAV	Encoded access mode for MPUI port. 00: Shared access MPUI port RAM, shared access peripheral 01: Shared access MPUI port RAM, host only access peripheral 10: Host only access MPUI port RAM, shared access peripheral 11: Host only access MPUI port RAM, host only access peripheral	R	11
10:9	CS_SAV	Indicates the transaction chip-select on abort. 01: MPUI port memory chip-select 10: DSP core TIPB peripheral or MPUI port control register (depending upon address value for the aborted transaction)	R	00

Table 102. Debug Flag Register (DEBUG_FLAG) (Continued)

Base Address = 0xFFFE C900, Offset = 0x0C				
Bit	**Name**	**Function**	**R/W**	**Reset**
8:6	BURST_SIZE_SAV	System bus data burst size indicated from the MPU core on abort. Can be used in conjunction with BURST_SIZE_ERR flag bit. Valid OMAP 3.2 burst size values: 000: One data burst (single-access) 011: Four-data burst Since the MPUI port only supports single-access, a 000 value is expected; otherwise, an abort is generated.	R	000
5	RNW_SAV	Indicates read or write transaction on the MPUI upon abort. 0: Write transaction. 1: Read transaction.	R	0
4	BYTE_SAV	Data width of the MPUI port access upon abort. Note that the MPUI handles 32-bit accesses from MPU core/OCP-I/system DMA as successive 16-bit transactions to fit the 16-bit interface of the MPUI port. 0: 8-bit access. 1: 16-bit access.	R	0
3	BURST_SIZE_ERR	Flag set to 1 when the system bus data burst size indicated from the MPU core is not 000. Because MPUI port supports single-access only, a 000 value is always expected. The value of the burst indicated by MPU core is saved in bits (8:6). When read, this bit is reset to 0.	R	0
2	TIMEOUT_ERR	Flag set to 1 when MPUI access is aborted by internal timeout. When read, this bit is reset to 0.	R	0
1	API_ERR	Flag set to 1 when MPUI port aborts access. When read, this bit is reset to 0.	R	0
0	ABORT_FLAG	Flag set to 1 when MPUI access is aborted. When read, this bit is reset to 0.	R	0

Table 103. MPUI Status Register (MPUI_STATUS)

Base Address = 0xFFFE C900, Offset = 0x10

Bit	Name	Function	R/W	Reset
31:13	Reserved		R/W	0x0000
12:11	ACCESS_STATUS	Current access in progress is: 00: MPU core access 01: System DMA access 10: OCP-I access 11: No access	R	11
10:3	TIMEOUT_VAL	Current value of timeout counter	R	0xFF
2	CS_EN	MPUI busy status. 0: MPUI is busy executing a transaction or host/shared mode switch. All new MPU core/OCP-I/system DMA accesses must wait until CS_EN is 1. 1: A new access may be started in the MPUI.	R	1
1	ACCESS_DONE	MPUI access status (similar to CS_EN, without mode switch indication). 0: MPUI is accessing MPUI port. 1: No access in progress; last access is completed.	R	1
0	HOMNSAM_FLAG	Current access mode when ACCESS_DONE = 0, or last access mode when ACCESS_DONE = 1. 0: SAM. 1: HOM.	R	1

Table 104 describes (DSP_STATUS) bits. For debug purposes, this register stores the state of several signals internal to the DSP core subsystem. Values are captured at each MPUI clock cycle.

Table 104. DSP Core Status Register (DSP_STATUS)

Base Address = 0xFFFE C900, Offset = 0x14				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:12	Reserved		R	0x0000
11	HRHOMNSAM	Reflects DSP core HOM or SAM setting for DSP core TIPB.	R	0
10	HAHOMNSAM	Reflects DSP core HOM or SAM setting for MPUI port RAM.	R	0
9	PENRESETDPLL	Reflects level of asynchronous reset in the DSP core (controlled by emulation).	R	0
8	PEIDLE7	Idle peripherals flag. Reflects bit 7 of (ISTR) from the DSP core.	R	0
7	PEIDLE6	Idle peripherals flag. Reflects bit 6 of (ISTR) from the DSP core.	R	0
6	PEIDLEDPLL	Idle DPLL flag. Reflects bit 4 of (ISTR) from the DSP core.	R	0
5	PEIDLEPERIPH	Idle peripherals flag. Reflects bit 3 of (ISTR) from the DSP core.	R	0
4	CPUIACK	Reflects level of Interrupt acknowledged signal from the DSP core.	R	0
3	CPUAVIS	Reflects bit 4 from (DSP_STATUS).	R	0
2	CPUXF	Reflects level of XF output from (DSP_STATUS).	R	0
1	RESET_MCU	Reflects level of the secondary DSP core subsystem reset originating from the MPU core (active low). Signal used to reset the DSP core TIPB interrupt priority encoder, the EMIF configuration registers, and the MPUI port control logic.	R	0
0	RESET	Reflects level of DSP core subsystem master reset (active low). Signal used to reset the entire DSP core subsystem except for the DSP core TIPB interrupt priority encoder, the EMIF configuration registers, and the MPUI port control logic.	R	0

Table 105. DSP Core Boot Configuration Register (DSP_BOOT_CONFIG)

Base Address = 0xFFFE C900, Offset = 0x18				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:16	Reserved		R/W	0x0000
15:10	BOOT_RHEA_PTR2	User-defined pointer that can be used for application-specific boot code location.	R/W	000000
9:4	BOOT_RHEA_PTR1	User-defined pointer that can be used for application-specific boot code location.	R/W	000000
3:0	DSP_BOOT_MODE	DSP core boot mode inputs.	R/W	0000

Table 106. MPUI Port RAM Configuration Register (DSP_MPUI_CONFIG)

Base Address = 0xFFFE C900, Offset = 0x1C				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:16	Reserved		R/W	0x0000
15:0	API_SIZE	Grants the MPUI exclusive access to the specified portion of DSP core SARAM in HOM. For details on SARAM configuration, see Table 109.	R/W	0xFFFF

Table 107 describes the DSP core miscellaneous register bits. For debug purposes, this register stores the state of the BION signal (internal to the DSP core subsystem). State is captured at each MPUI clock cycle.

Table 107. DSP Core Miscellaneous (DSP_MISC)

Base Address = 0xFFFE C900, Offset = 0x20				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:9	Reserved		R/W	0x0000
8	CPUBION	Reflects level of BION signal to DSP core subsystem.	R	0
7:0	Reserved		R/W	0x00

Table 108. MPUI Enhanced Control Register (MPUI_ENHANCED_CTRL)

Base Address = 0xFFFE C900, Offset = 0x24				
Bit	**Name**	**Function**	**R/W**	**Reset**
31:1	Reserved		R/W	0x0000
0	DPS_EN	When 1, dynamic power saving (DPS) mode is enabled. Otherwise, DPS is disabled. When DPS is enabled, the MPUI clock is turned off when there is no active request from the MPU core/DSP DMA/System DMA/OCPI.	R/W	0

Table 109. DSP Core SARAM Configuration Map

API SIZE	SARAM 28	SARAM 24	SARAM 20	SARAM 16	SARAM 12	SARAM 8	SARAM 4	SARAM 0
0x0000−0x0001	0000	0000	0000	0000	0000	0000	0000	0000
0x0002−0x0003	0000	0000	0000	0000	0000	0000	0000	0001
0x0004−0x0005	0000	0000	0000	0000	0000	0000	0000	0011
0x0006−0x0007	0000	0000	0000	0000	0000	0000	0000	0111
0x0008−0x0009	0000	0000	0000	0000	0000	0000	0000	1111
0x000A−0x000B	0000	0000	0000	0000	0000	0000	0001	1111
0x000C−0x000D	0000	0000	0000	0000	0000	0000	0011	1111
0x000E−0x000F	0000	0000	0000	0000	0000	0000	0111	1111
0x0010−0x0011	0000	0000	0000	0000	0000	0000	1111	1111
0x0012−0x0013	0000	0000	0000	0000	0000	0001	1111	1111
0x0014−0x0015	0000	0000	0000	0000	0000	0011	1111	1111
0x0016−0x0017	0000	0000	0000	0000	0000	0111	1111	1111
0x0018−0x0019	0000	0000	0000	0000	0000	1111	1111	1111
0x001A−0x001B	0000	0000	0000	0000	0001	1111	1111	1111
0x001C−0x001D	0000	0000	0000	0000	0011	1111	1111	1111
0x001E−0x001F	0000	0000	0000	0000	0111	1111	1111	1111
0x0020−0x0021	0000	0000	0000	0000	1111	1111	1111	1111
0x0022−0x0023	0000	0000	0000	0001	1111	1111	1111	1111
0x0024−0x0025	0000	0000	0000	0011	1111	1111	1111	1111
0x0026−0x0027	0000	0000	0000	0111	1111	1111	1111	1111

Table 109. DSP Core SARAM Configuration Map (Continued)

API SIZE	SARAM 28	SARAM 24	SARAM 20	SARAM 16	SARAM 12	SARAM 8	SARAM 4	SARAM 0
0x0028−0x0029	0000	0000	0000	1111	1111	1111	1111	1111
0x002A−0x002B	0000	0000	0001	1111	1111	1111	1111	1111
0x002C−0x002D	0000	0000	0011	1111	1111	1111	1111	1111
0x002E−0x002F	0000	0000	0111	1111	1111	1111	1111	1111
0x0030−0x0031	0000	0000	1111	1111	1111	1111	1111	1111
0x0032−0x0033	0000	0001	1111	1111	1111	1111	1111	1111
0x0034−0x0035	0000	0011	1111	1111	1111	1111	1111	1111
0x0036−0x0037	0000	0111	1111	1111	1111	1111	1111	1111
0x0038−0x0039	0000	1111	1111	1111	1111	1111	1111	1111
0x003A−0x003B	0001	1111	1111	1111	1111	1111	1111	1111
0x003C−0x003D	0011	1111	1111	1111	1111	1111	1111	1111
0x003E−0x003F	0111	1111	1111	1111	1111	1111	1111	1111
0x0040−OTHERS	1111	1111	1111	1111	1111	1111	1111	1111

This table decodes the API_SIZE value into the exclusively accessible portion of SARAM. The SARAM has 32 blocks (SARAM0 through SARAM31) on 2 KB boundaries. The exclusively accessible memory (host-only RAM) is marked with a 1, and nonexclusively accessible memory (shared access RAM) is marked with a 0.

5.4 DSP Core Endianism Register

Note that this register is part of the Traffic Controller block.

Table 110. DSP Core Endianism Register (DSP_ENDIAN_CONV)

Address = 0xFFFE CC34

Bit	Name	Function	R/W	Reset
31:2	Reserved	Reserved.	R	ND
1	SWAP	0: Byte swap. 1: Word swap.	R/W	0
0	EN	0: Disables DSP core Endianism conversion. 1: Enables DSP core Endianism conversion.	R/W	0

This register controls endianism for all DSP core accesses through the traffic controller.

6 Mailboxes

6.1 Mailbox Registers

- ❑ The MPU core/DSP DMA/System DMA/OCP-I uses the following registers to communicate with the DSP core:
  - MPU2DSP1A and MPU2DSP1B for mailbox1
  - MPU2DSP2A and MPU2DSP2B for mailbox2

  The sequence is as follows:

  - An interrupt for the DSP core is generated when the MPU core/DSP DMA/System DMA/OCP-I writes to the second register (MPU2DSP1B or MPU2DSP2B). Writing to the first register (MPU2DSP1A or MPU2DSP2A) does not generate an interrupt.
  - Once an interrupt has been set up, registers associated with this mailbox interrupt are locked until DSP core clears the interrupt flag by reading the second register. For example, when MPU2DSP1B is written by the MPU core, the MPU core can write to neither MPU2DSP1A or MPU2DSP1B until the DSP core clears the interrupt flag by reading MPU2DSP1B.
- ❑ The DSP core uses the following registers to communicate with the MPU core:
  - DSP2MPU1A and DSP2MPU1B for mailbox1
  - DSP2MPU2A and DSP2MPU2B for mailbox2

6.1.1 Mailbox Interrupts

Each mailbox interrupt is also associated to an interrupt flag bit (INT) which is set to 1 when the interrupt is pending. The four flag registers are as follows:

- ❑ MPU core flag registers:
  - MPU2DSP1_FLAG
  - MPU2DSP2_FLAG
- ❑ DSP flag registers:
  - DSP2MPU1_FLAG
  - DSP2MPU2_FLAG

These flag registers reflect only the state of the mailbox; a write into them has no effect.

The interrupt flag register is set to 1 when a write is detected in the second register of a mailbox. This interrupt flag is cleared when a read is detected on the second data register, by the interrupted processor of a mailbox interrupt (see section 6.2 for details).

When the interrupt flag has been set up by a processor, this processor can write to neither of the registers associated with that mailbox interrupt until the other processor cleans the interrupt flag.

For example, if the MPU core sets MPU2DSP1_FLAG by writing to MPU2DSP1B, the MPU core has no write access to MPU2DSP1A or MPU2DSP1B until the DSP core clears the interrupt flag by reading the MPU2DSP1B.

Because a processor cannot write to registers associated with an interrupt after the interrupt flag has been set up, both processors must write the data or command that needs to be communicated before setting the interrupt flag register. For example, MPU2DSP1A must be written to before MPU2DSP1B, and so on.

If the interrupt flag is not cleared and a processor tries to write to the associated mailbox register, then the write is ignored. Also, the TIPB bridge generates an abort (time-out abort), and the program returns to the processor to continue its normal mode of execution. Even though the interrupt flag is not cleared, the processor can read its own registers.

6.1.2 Mailbox Software

To use the mailbox to communicate from the MPU core to the DSP core, program the mailbox registers so the following sequence of steps occurs in sequence. The programmed sequence to communicate from the DSP core to the MPU core is the same.

1) The MPU core enables the MPU2DSP1 mailbox interrupt and configures the mailbox interrupt as level-sensitive in the interrupt handlers.

2) The MPU core writes to MPU2DSP1A and MPU2DSP1B.

3) Writing to MPU2DSP1B sets MPU2DSP1_FLAG.INT = 1 and generates an interrupt toward the DSP core.

4) The MPU core can now poll MPU2DSP1_FLAG. As long as this register is set (it has a value of 1), the DSP core has not cleared the interrupt and the MPU core cannot generate the next mailbox interrupt.

5) The DSP core services this interrupt in an interrupt service routine. MPU2DSP1_FLAG is cleared when the DSP core reads from MPU2DSP1B. Note that reading by the MPU core does not clear MPU2DSP1_FLAG.

6) The MPU core can generate the next interrupt by writing MPU core-to-DSP core mailbox registers, after MPU2DSP1_FLAG has been cleared.

Masking a mailbox interrupt does not disable that interrupt. If a mailbox interrupt is generated while it is masked, it remains pending until the mask is removed, when the interrupt becomes an active interrupt. To clear a masked interrupt, appropriate mailbox registers must be read.

6.2 Registers

All these registers are 16-bit aligned on a 32-bit address boundary. The DSP core-to-MPU core mailbox registers are written by the DSP core and read by the MPU core/DSP DMA/System DMA/OCP-I, while the MPU core-to-DSP core mailbox registers are written by the MPU core/DSP DMA/System DMA/OCP-I and read by the DSP core. Table 111 lists the mailbox registers. Table 112 through Table 123 provide register bit descriptions.

Table 111. Mailbox Registers

Base Address = 0xFFFC F000			
Name	**Description**	**R/W**	**Offset**
MPU2DSP1A	MPU to DSP mailbox 1A	R/W	0x00
MPU2DSP1B	MPU to DSP mailbox 1B	R/W	0x04
DSP2MPU1A	DSP to MPU mailbox 1A	R/W	0x08
DSP2MPU1B	DSP to MPU mailbox 1B	R/W	0x0C
DSP2MPU2A	DSP to MPU mailbox 2A	R/W	0x10
DSP2MPU2B	DSP to MPU mailbox 2B	R/W	0x14
MPU2DSP1_FLAG	MPU to DSP mailbox 1 flag	R	0x18
DSP2MPU1_FLAG	DSP to MPU mailbox 1 flag	R	0x1C
DSP2MPU2_FLAG	DSP to MPU mailbox 2 flag	R	0x20
MPU2DSP2A	MPU to DSP mailbox 2A	R/W	0x24
MPU2DSP2B	MPU to DSP mailbox 2B	R/W	0x28
MPU2DSP2_FLAG	MPU to DSP mailbox 2 flag register	R	0x2C

Table 112. MPU to DSP Mailbox 1A Register (MPU2DSP1A)

Base Address = 0xFFFC F000, Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	MPU2DSP1A	This register stores the data to be shared for the MPU-to-DSP interrupt in mailbox 1.	R/W by MPU core/DSP DMA/System DMA/OCP-I R by DSP core	0x0000

Table 113. MPU to DSP Mailbox 1B Register (MPU2DSP1B)

Base Address = 0xFFFC F000, Offset = 0x04				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	MPU2DSP1B	This register stores the data to be shared for the MPU-to-DSP interrupt in mailbox 1. The MPU2DSP1 interrupt is generated to DSP core when this register is written. When this register is read by DSP core, (MPU2DSP1_FLAG) is reset.	R/W by MPU core/DSP DMA/System DMA/OCP-I R by DSP core	0x0000

Table 114. DSP to MPU Mailbox 1A Register (DSP2MPU1A)

Base Address = 0xFFFC F000, Offset = 0x08				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	DSP2MPU1A	This register stores the data to be shared for the DSP-to-MPU interrupt in mailbox 1.	R/W by DSP core R by MPU core/DSP DMA/System DMA/OCP-I	0x0000

Table 115. DSP to MPU Mailbox 1B Register (DSP2MPU1B)

Base Address = 0xFFFC F000, Offset = 0x0C				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	DSP2MPU1B	This register stores the data to be shared for the DSP-to-MPU interrupt in mailbox 1. The DSP2MPU1 interrupt is generated to MPU core/DSP DMA/System DMA/OCP-I when this register is written. When this register is read by MPU core, (DSP2MPU1_FLAG) is reset.	R/W by DSP core R by MPU core/DSP DMA/System DMA/OCP-I	0x0000

Table 116. DSP to MPU Mailbox 2A Register (DSP2MPU2A)

Base Address = 0xFFFC F000, Offset = 0x10				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	DSP2MPU2A	This register stores the data to be shared for the DSP-to-MPU interrupt in mailbox 2.	R/W by DSP core R by MPU core/DSP DMA/System DMA/OCP-I	0x0000

Table 117. DSP to MPU Mailbox 2B Register (DSP2MPU2B)

Base Address = 0xFFFC F000, Offset = 0x14				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	DSP2MPU2B	This register stores the data to be shared for the DSP-to-MPU interrupt in mailbox 2. The DSP2MPU2 interrupt is generated to MPU core/DSP DMA/System DMA/OCP-I when this register is written. When this register is read by MPU core, (DSP2MPU2_FLAG) is reset.	R/W by DSP core R by MPU core/DSP DMA/System DMA/OCP-I	0x0000

Table 118. MPU to DSP Mailbox 1 Flag Register (MPU2DSP1_FLAG)

Base Address = 0xFFFC F000, Offset = 0x18				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:1	Reserved		R/W	0x0000
0	INT	0: No interrupt pending. 1: Interrupt generated.	R by MPU core/DSP DMA/System DMA/OCP-I No access by DSP core	0

Table 119. DSP to MPU Mailbox 1 Flag Register (DSP2MPU1_FLAG)

Base Address = 0xFFFC F000, Offset = 0x1C				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:1	Reserved		R/W	0x0000
0	INT	0: No interrupt pending. 1: Interrupt generated.	R by DSP core No access by MPU core/DSP DMA/System DMA/OCP-I	0

Table 120. DSP to MPU Mailbox 2 Flag Register (DSP2MPU2_FLAG)

Base Address = 0xFFFC F000, Offset = 0x20				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:1	Reserved		R/W	0x0000
0	INT	0: No interrupt pending. 1: Interrupt generated.	R by DSP core No access by MPU core/DSP DMA/System DMA/OCP-I	0

Table 121. MPU to DSP Mailbox 2A Register (MPU2DSP2A)

Base Address = 0xFFFC F000, Offset = 0x24				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	MPU2DSP2A	This register stores the data to be shared for the MPU-to-DSP interrupt in mailbox 2.	R/W by MPU core/DSP DMA/System DMA/OCP-I R by DSP core	0x0000

Table 122. MPU to DSP Mailbox 2B Register (MPU2DSP2B)

Base Address = 0xFFFC F000, Offset = 0x28				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	MPU2DSP2B	This register stores the data to be shared for the MPU-to-DSP interrupt in mailbox 2. The MPU2DSP2 interrupt is generated to DSP core when this register is written. When this register is read by DSP core, (MPU2DSP2_FLAG) is reset.	R/W by MPU core/DSP DMA/System DMA/OCP-I R by DSP core	0x0000

Table 123. MPU to DSP Mailbox 2 Flag Register (MPU2DSP2_FLAG)

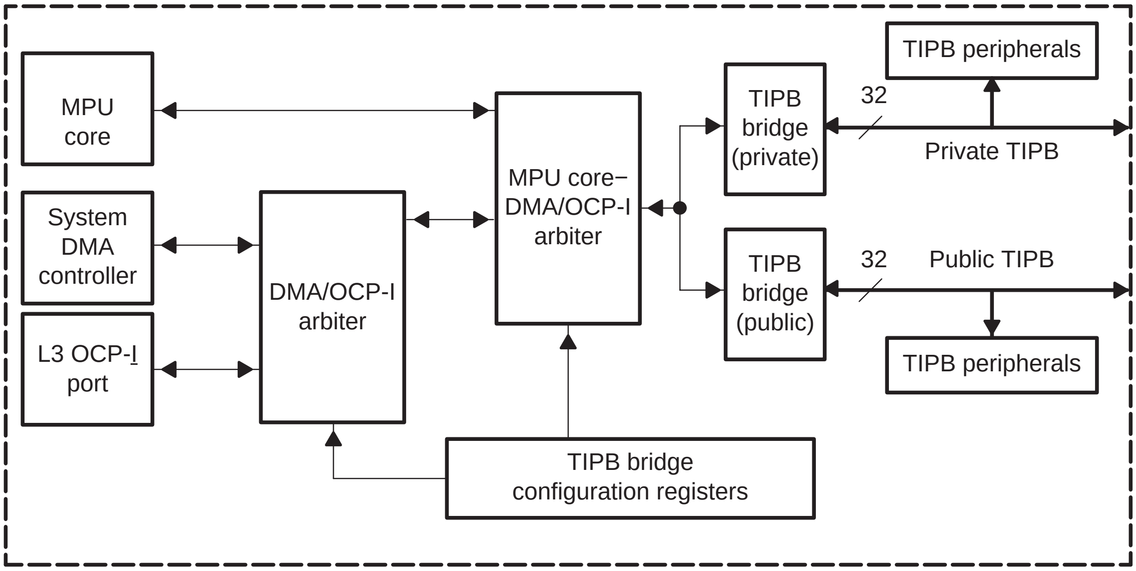
Base Address = 0xFFFC F000, Offset = 0x2C				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:1	Reserved	Reserved.	R	0
0	INT	0: No interrupt pending. 1: MPU2DSP2 interrupt generated.	R by MPU core/DSP DMA/System DMA/OCP-I No access by DSP core	0

7 TIPB Bridge

The TIPB bridge allows the TIPB and all connected peripherals to be shared with three hosts: the MPU core, the OCP initiator (OCP-I), and the system DMA controller. The TIPB controls accesses to avoid conflicts between the hosts, and it allows the MPU core to configure the protocol parameters of the TIPB.

The TIPB bridge module is shown in Figure 43.

Figure 43. OMAP 3.2 Platform TI Peripheral Bridge



7.1 Functionality

This section describes the functionality of the TI peripheral bus bridge.

7.1.1 Bus Allocation

The TIPB is shared between the MPU core memory interface, the OCP initiator (OCP-I), and the DMA controller. Two levels of bus allocation are used to resolve conflicts and prioritize accesses among the three requestors.

The first level of the two-level arbitration process selects between the system DMA and OCP-I for control of the TIPB. Fixed or round-robin priority schemes can be programmed in the FIXNROUND_PRIORITY bit field of the TIPB allocation control register, (RHEA_BUS_ALLOC). When fixed priority is selected, the EXTNINT_PRIORITY bit field determines whether system DMA or OCP-I has fixed priority.

The second level of the two-level arbitration process selects between the MPU core and the system DMA/OCP-I. The value programmed in the RHEA_PRIORITY bits of (RHEA_BUS_ALLOC) defines the priority. If the value is 0, the MPU core memory interface has priority over the system DMA/OCP-I. If the value equals *n* (where *n* is from 1 to 7), the system DMA/OCP-I has priority over the MPU core and it can perform *n* accesses before giving the priority back to MPU core memory interface.

7.1.2 Access Permissions

Access permissions to the MPU core public and private TIPBs vary with initiator and target.

The MPU core has unconditional access permissions to the MPU core public and private TIPBs.

The system DMA has unconditional access permission to the MPU core public TIPB and conditional, software-controlled access permission to the MPU core private TIPB. System DMA permission for the MPU core private TIPB is programmed in (ACCESS_CNTL).

The OCP-I has conditional, software-controlled access to both the MPU core public and private TIPBs. Refer to the *Traffic Controller* section for details on programming OCP-I access permissions.

7.1.3 TIPB Strobes and Access Factor

The TIPB strobes are an integral part of the TIPB bridge. The TIPB strobes are (active low) signals output from the TIPB bridge module that drive the peripheral interfaces of the MPU core TIPB. The TIPB bridge uses two strobes (strobe 0 and strobe 1), and each strobe is fixed to control access to distinct ranges of the MPU core TIPB peripherals address space. In addition, address spaces defined to strobe 0 and strobe 1 are further segmented into areas of MPU core public TIPB and MPU core private TIPB memory space. Refer to the MPU core Peripherals Memory Mapping section in the *OMAP5912 Data Manual* (SPRS231)*,* for details on strobe address space assignments.

The TIPB bridge may be required to communicate with peripherals of varying speeds. To allow slow peripherals to answer, it is possible to lengthen the strobe 0 and strobe 1 periods using ACCESS_FACTOR0 and ACCESS_FACTOR1 in (RHEA_CNTL). By programming ACCESS_FACTOR0 and ACCESS_FACTOR1 to *n*, the respective strobe stretches its access over 2**n* TIPB bridge clock cycles (*n* cycles the strobe is inactive high; *n* cycles the strobe is active low).

The TIPB bridge clock referenced here is the input clock reference to the TIPB bridge module generated from the clock and reset management module. The clock rate for the TIPB bridge is fixed to the same value as that of the OMAP 3.2 traffic controller module. For details, see the *OMAP5912 Multimedia Processor Clocks Reference Guide* (SPRU751).

7.1.4 MPU Core Posted Write

The MPU core can perform a posted write. When posted write is enabled inside (ARM_RHEA_CNTL), data sent by the MPU core is buffered in the TIPB bridge module, and the MPU core can continue accessing other locations. The bridge handles the access to the TIPB peripheral so that the MPU core is not stalled during the access.

If the MPU core/system DMA/OCP-I performs another TIPB operation when there is a posted write, this operation must wait until the posted write is complete. If the system DMA or OCP-I performs a read operation to the same address as the posted write, the posted write data is not forwarded to the system DMA or OCP-I. Posted write is not supported for system DMA and OCP-I accesses.

Using (ARM_RHEA_CNTL), posted write can be enabled independently for strobe 0 or strobe 1 address spaces. This provides some flexibility over which MPU core TIPB peripherals are configured for posted write. Refer to the memory map in the *OMAP5912 Data Manual* (SPRS231) for details on strobe address space assignments.

7.1.5 Time-Out

A TIPB access time-out limits the maximum time a peripheral can stall the processor. When starting an access on the TIPB, the time-out counter is loaded with the value programmed in the TIMEOUT bits of (RHEA_CNTL). If the current access is not finished when the counter reaches 0, the cycle is aborted and an abort is generated to the MPU core/system DMA/OCP-I.

The time-out value is calculated by:

$t_{\text{time-out}} = (1/f_{\text{bridge\_clk}}) \times ((\text{RHEA\_CNTL.TIMEOUT}) + 1)$

where $f_{\text{bridge\_clk}}$ is the traffic controller clock frequency setting.

The time-out can be used in conjunction with the posted write. The counter begins counting down when the posted write transaction has been scheduled, and this count continues against the posted transaction even if another transaction to the TIPB bridge occurs.

The time-out can be disabled using the TIMEOUT_EN bit in (ENH_RHEA_CNTL).

7.1.6 Debug

Debug registers are saved when an MPU_TIPB abort occurs. Abort can be caused by time-out, or by size mismatch between the access word width and the word width of the addressed peripheral.

The access address, data and error flags are saved to (DEBUG_ADDRESS), (DEBUG_DATA_LSB), (DEBUG_DATA_MSB), and (DEBUG_CTRL_SIGNALS).

7.2 Registers

All TIPB bridge configuration and debug registers are 16-bit registers. Write accesses to all TIPB registers can be performed only in MPU core supervisor mode. Read accesses can be performed in MPU core supervisor or user modes.

Table 124 provides a list of the TIPB registers. Table 125 through Table 133 provide register bit descriptions.

Table 124. TIPB Registers

Base Address = 0xFFFE D300 (public), 0xFFFE CA00 (private)			
Name	**Description**	**R/W**	**Offset**
RHEA_CNTL	TIPB control		0x00
RHEA_BUS_ALLOC	TIPB allocation control		0x04
ARM_RHEA_CNTL	MPU core TIPB control		0x08
ENH_RHEA_CNTL	Enhanced TIPB control		0x0C
DEBUG_ADDRESS	Debug address		0x10
DEBUG_DATA_LSB	Debug data LSB		0x14
DEBUG_DATA_MSB	Debug data MSB		0x18
DEBUG_CTRL_SIGNALS	Debug control signals		0x1C
ACCESS_CNTL	Access control		0x20

Table 125. TIPB Control Register (RHEA_CNTL)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:8	TIMEOUT	TIPB bus access time-out. When starting an access on the TIPB bus, the time-out counter is loaded with this value. If the current access is not finished when the counter reaches 0, the cycle is aborted and abort indications are given to the peripheral and the MPU core, system DMA, or OCP-I. Maximum value for TIMEOUT is 255.	R/W	0xFF
7:4	ACCESS_FACTOR1	Clock period multiplication factor for TIPB strobe 1. Allows access to slow peripherals by lengthening the TIPB strobe 1 period by a multiple of the internal TIPB bridge clock period. Note that the TIPB bridge clock period is the same as the OMAP traffic controller clock period. 0000: Same as 0001 0001: Strobe period = TIPB bridge clock period x 2 0010: Strobe period = TIPB bridge clock period x 4 0011: Strobe period = TIPB bridge clock period x 6 0100: Strobe period = TIPB bridge clock period x 8 0101: Strobe period = TIPB bridge clock period x 10 0110: Strobe period = TIPB bridge clock period x 12 0111: Strobe period = TIPB bridge clock period x 14 1000: Strobe period = TIPB bridge clock period x 1001: Strobe period = TIPB bridge clock period x 18 1010: Strobe period = TIPB bridge clock period x 20 1011: Strobe period = TIPB bridge clock period x 22 1100: Strobe period = TIPB bridge clock period x 24 1101: Strobe period = TIPB bridge clock period x 26 1110: Strobe period = TIPB bridge clock period x 28 1111: Strobe period = TIPB bridge clock period x 30	R/W	0x1

Table 125. TIPB Control Register (RHEA_CNTL) (Continued)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x00				
Bit	**Name**	**Function**	**R/W**	**Reset**
3:0	ACCESS_FACTOR0	Clock period multiplication factor for TIPB strobe 0. Allows access to slow peripherals by lengthening the TIPB strobe 0 period by a multiple of the internal TIPB bridge clock period. Note that the TIPB bridge clock period is the same as the OMAP traffic controller clock period. 0000: Same as 0001. 0001: Strobe period = TIPB bridge clock period x 2 0010: Strobe period = TIPB bridge clock period x 4 0011: Strobe period = TIPB bridge clock period x 6 0100: Strobe period = TIPB bridge clock period x 8 0101: Strobe period = TIPB bridge clock period x 10 0110: Strobe period = TIPB bridge clock period x 12 0111: Strobe period = TIPB bridge clock period x 14 1000: Strobe period = TIPB bridge clock period x 1001: Strobe period = TIPB bridge clock period x 18 1010: Strobe period = TIPB bridge clock period x 20 1011: Strobe period = TIPB bridge clock period x 22 1100: Strobe period = TIPB bridge clock period x 24 1101: Strobe period = TIPB bridge clock period x 26 1110: Strobe period = TIPB bridge clock period x 28 1111: Strobe period = TIPB bridge clock period x 30	R/W	0x1

Table 126. TIPB Allocation Control Register (RHEA_BUS_ALLOC)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x04				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:6	Reserved		R/W	0x000
5	EXTNINT_PRIORITY	Priority between system DMA and OCP-I when fixed priority scheme is selected. 0: System DMA has priority. 1: OCP-I has priority.	R/W	0

Table 126. TIPB Allocation Control Register (RHEA_BUS_ALLOC) (Continued)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x04				
Bit	**Name**	**Function**	**R/W**	**Reset**
4	FIXNROUND_PRIORITY	Type of priority scheme used in system DMA and OCP-I arbitration: 0: Round-robin scheme used. 1: Fixed priority scheme used.	R/W	0
3	PRIORITY_ENABLE	0: TIPB bus allocation is done using the RHEA_PRIORITY bits. 1: MPU core has the same priority as the system DMA/OCP-I transfers regarding TIPB bus allocation when it is in exception mode (IRQ and FIQ).	R/W	1
2:0	RHEA_PRIORITY	Defines TIPB priority between MPU core and system DMA/OCP-I. 000: MPU core has priority over the system DMA/OCP-I. 001 through 111: system DMA/OCP-I has priority over the MPU core, and can perform the programmed number of accesses before the MPU core can access the bus.	R/W	001

Table 127. MPU Core TIPB Control Register (ARM_RHEA_CNTL)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x08				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:2	Reserved		R/W	0x0000
1	W_BUF_EN_1	0: Posted write buffer is bypassed. 1: Posted write buffer is enabled for MPU core public and private TIPB peripherals that have address space assigned to TIPB strobe 1.	R/W	0
0	W_BUF_EN_0	0: Posted write buffer is bypassed. 1: Posted write buffer is enabled for MPU core public and private TIPB peripherals that have address space assigned to TIPB strobe 0.	R/W	0

Table 128. Enhanced TIPB Control Register (ENH_RHEA_CNTL)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x0C				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:4	Reserved		R/W	0x000
3	MASK_ABORT	0: An abort signal is sent to the MPU core whenever an MPU to_TIPB access is aborted. 1: The abort signal is not sent. MASK_ABORT does not mask/unmask system DMA or OCP-I aborts.	R/W	1
2	Not Used	Not used in OMAP 3.2 (always one). Legacy HIGH_FREQ mode from OMAP 3.0/3.1.		1
1	MASK_IT	0: An interrupt is sent to the MPU core whenever a TIPB write access (from MPU core/system DMA/OCP-I) is aborted or any TIPB access has a size mismatch. 1: The interrupt is masked.	R/W	1
0	TIMEOUT_EN	0: Do not enable the TIMEOUT feature. 1: Enable the TIMEOUT feature.	R/W	1

Table 129. Debug Address Register (DEBUG_ADDRESS)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x10				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	ADDRESS_DBG	Address from MPU core memory interface; saved when an abort or access size mismatch occurs.	R	0xFFFF

Table 130. Debug Data LSB Register (DEBUG_DATA_LSB)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x14				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	DATA_DBG_LOW	Bits 15 to 0 of data bus from MPU core. The value of the MPU core data input is saved when a read access has a size mismatch. The MPU core data output bus is saved when a write access is aborted or has a size mismatch. If a read access is aborted, the value of this register is irrelevant.	R	0xFFFF

Table 131. Debug Data MSB Register (DEBUG_DATA_MSB)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x18				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:0	DATA_DBG_HIGH	Bits 31 to 16 of data bus from MPU core. The value of the MPU core data input bus is saved when a read access has a size mismatch. The MPU core data output bus is saved when a write access is aborted or has a size mismatch. If a read access is aborted, the value of this register is irrelevant.	R	0xFFFF

Table 132. Debug Control Signals Register (DEBUG_CTRL_SIGNALS)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x1C				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:11	Reserved		R/W	0x00
10:9	HOST_ID	Host-ID that caused the abort: 00: MPU core 01: System DMA 10: OCP-I 11: Invalid	R	00
8	BURST_ACC	Indicates single or burst access on the TIPB; saved when abort or access size mismatch occurs. 0: Single access. 1: Burst access.	R	0
7:6	DBG_PERHMAS(1:0)	Peripheral memory access size on TIPB; saved when abort or access size mismatch occurs. 00: 8 bits 01: 16 bits 1x: 32 bits	R	11
5:4	DBG_MAS(1:0)	Memory access size on TIPB; saved when abort or access size mismatch occurs. 00: 8 bits 01: 16 bits 1x: 32 bits	R	11

Table 132. Debug Control Signals Register (DEBUG_CTRL_SIGNALS) (Continued)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x1C				
Bit	**Name**	**Function**	**R/W**	**Reset**
3	DBG_NSUPV	Indicates supervisor mode status of MPU core; saved when abort or access size mismatch occurs. 0: Processor in supervisor mode. 1: Processor not in supervisor mode.	R	1
2	DBG_RNW	Indicates read or write transaction on the TIPB; saved when abort or access size mismatch occurs. 0: Write transaction. 1: Read transaction.	R	1
1	WR_SIZE_FLAG	Flag set to 1 when there is a mismatch between memory access size and peripheral memory access size. When read, the bit is reset to 0.	R	0
0	ABORT_FLAG	Flag set to 1 when TIPB access is aborted. When read, the bit is reset to 0.	R	0

Table 133. Access Control Register (ACCESS_CNTL)

Base Address = 0xFFFE D300 (Public), 0xFFFE CA00 (Private), Offset = 0x20				
Bit	**Name**	**Function**	**R/W**	**Reset**
15:4	Reserved	Reserved.	R/W	0x000
3	DPS_EN	0: Dynamic power-saving mode is disabled. 1: Dynamic power-saving mode is enabled. When DPS is enabled, the bridge clock is turned.	R/W	0
2	MASK_OCPI NABORT	0: The abort for system DMA access is sent back to the system DMA. 1: The abort for OCPI access is masked before sending back to the OCPI.	R/W	0
1	MASK_DMA NABORT	1: The abort for system DMA access is masked before sending back to the system DMA. 0: The abort for system DMA access is sent back to the system DMA.	R/W	0
0	DMA_ENABLE	1: System DMA can access peripherals on the TIPB bridge 0: System DMA cannot access peripherals on the TIPB bridge	R/W	1

This page is intentionally left blank.

Revision History

Table 134 lists the changes made since the previous version of this document.

Table 134. Document Revision History

Page	Additions/Modifications/Deletions
84	Added a bullet to Section 3.5
76	Replaced Table 5
95	Changed bits 11−8 from reserved to WRRDYMASK_CSn in the EMIFS Dynamic Wait States Control Register (EMIFS_DWS) and added a paragraph preceding Table 24
97	Added Table 25 Read and Write Access Handshaking
98	Changed bits 10 and 11 in Table 29 from reserved to CLKMASK and READY_CONFIG
99	Added the EMIFF_DOUBLER register to Table 30
100	Added Table 31 for the EMIFF TC Doubler register
101	Changed Table 32 header to Base Address = 0xFFFE CC00, Offset = 0x08
105	Added Table 37 AC Timing When NEW_SYS_FREQ (bit 3 of EMIFF_CONFIG2) = 0
106	Added Table 38 AC Timing When NEW_SYS_FREQ (bit 3 of EMIFF_CONFIG2) = 1
107	Changed bit 4 from reserved to NEW_SYS_FREQ in Table 40
117	Added the OCPI Dynamic Power Down register to Table 58
120	Added Section 3.5.5
120	Added paragraph of text and Table 64 to describe the Dynamic Power Down register
121	Added Table 66 Reserved Space Fault Type Register (RES_SPC_ATYPER)
121	Added Table 67 Reserved Space Fault Address Register (RES_SPC_ADDR)