TMS320C674x/OMAP-L1x Processor External Memory Interface B (EMIFB)

User's Guide

Literature Number: SPRUFL7A
April 2009

Contents

List of Figures

List of Tables

Read This First

About This Manual

This document describes the operation of the external memory interface B (EMIFB).

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.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.

Related Documentation From Texas Instruments

The following documents describe the TMS320C674x Digital Signal Processors (DSPs) and OMAP-L1x Applications Processors. Copies of these documents are available on the Internet at www.ti.com. *Tip:* Enter the literature number in the search box provided at www.ti.com.

The current documentation that describes the DSP, related peripherals, and other technical collateral, is available in the C6000 DSP product folder at: www.ti.com/c6000.

SPRUGJ0 — *TMS320C6743 DSP System Reference Guide.* Describes the System-on-Chip (SoC) including the C6743 DSP subsystem, system memory, device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, and system configuration module.

SPRUFK4 — *TMS320C6745/C6747 DSP System Reference Guide.* Describes the System-on-Chip (SoC) including the C6745/C6747 DSP subsystem, system memory, device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, and system configuration module.

SPRUG84 — *OMAP-L137 Applications Processor System Reference Guide.* Describes the System-on-Chip (SoC) including the ARM subsystem, DSP subsystem, system memory, device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, ARM interrupt controller (AINTC), and system configuration module.

SPRUFK9 — *TMS320C674x/OMAP-L1x Processor Peripherals Overview Reference Guide.* Provides an overview and briefly describes the peripherals available on the TMS320C674x Digital Signal Processors (DSPs) and OMAP-L1x Applications Processors.

SPRUFK5 — *TMS320C674x DSP Megamodule Reference Guide.* Describes the TMS320C674x digital signal processor (DSP) megamodule. Included is a discussion on the internal direct memory access (IDMA) controller, the interrupt controller, the power-down controller, memory protection, bandwidth management, and the memory and cache.

SPRUFE8 — *TMS320C674x DSP CPU and Instruction Set Reference Guide.* Describes the CPU architecture, pipeline, instruction set, and interrupts for the TMS320C674x digital signal processors (DSPs). The C674x DSP is an enhancement of the C64x+ and C67x+ DSPs with added functionality and an expanded instruction set.

SPRUG82 — *TMS320C674x DSP Cache User's Guide.* Explains the fundamentals of memory caches and describes how the two-level cache-based internal memory architecture in the TMS320C674x digital signal processor (DSP) can be efficiently used in DSP applications. Shows how to maintain coherence with external memory, how to use DMA to reduce memory latencies, and how to optimize your code to improve cache efficiency. The internal memory architecture in the C674x DSP is organized in a two-level hierarchy consisting of a dedicated program cache (L1P) and a dedicated data cache (L1D) on the first level. Accesses by the CPU to the these first level caches can complete without CPU pipeline stalls. If the data requested by the CPU is not contained in cache, it is fetched from the next lower memory level, L2 or external memory.

External Memory Interface B (EMIFB)

1 Introduction

This section provides information about the purpose, and use of the external memory interface B (EMIFB). It also provides a block diagram of the EMIFB that shows its internal connections and external pins.

1.1 Purpose of the Peripheral

EMIFB memory controller is complaint with the JESD21-C SDR SDRAM memories utilizing either 32-bit or 16-bit of the EMIFB memory controller data bus. The purpose of this EMIFB is to provide a means for the CPU to connect to a variety of external devices including:

- Single data rate (SDR) SDRAM/ mobile SDR SDRAM

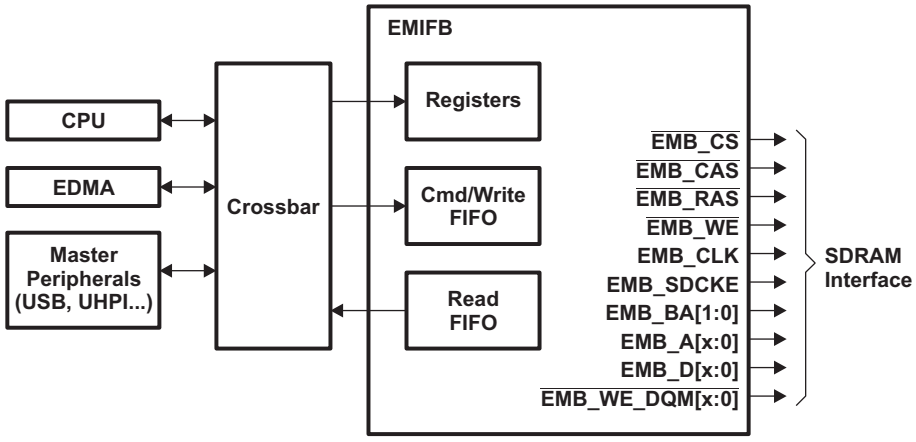
1.2 Features

For details on features of EMIFB, see your device-specific data manual.

1.3 Functional Block Diagram

Figure 1 illustrates a high-level view of the EMIFB and its connections within the device. Multiple requesters have access to EMIFB through a switched central resource(indicated as crossbar in the figure). The EMIFB implements a split transaction internal bus, allowing concurrence between reads and writes from the various requesters. Section 2.2 contains further description of the entities internal to the device that can send requests to the EMIFB. Section 2.3 describes the EMIFB external pins and summarizes their purpose when interfacing with SDRAM.

Figure 1. EMIFB Functional Block Diagram



2 Architecture

This section provides details about the architecture and operation of the EMIFB SDRAM interface.

2.1 Clock Control

For details on EMIFB clock control, see your device-specific *System Reference Guide*.

2.2 EMIF Requests

Depending on the specific device, different sources (CPU, EDMA, and other master peripherals) within the device can make requests to EMIFB. Some of these sources have multiple master ports to the crossbar (EDMA TPTCs) and some share ports to the crossbar (USB). The requests from these sources consist of accesses to SDRAM memory and EMIFB registers. The EMIFB implements internal data FIFOs and a split transaction internal bus to allow concurrence of read and write operations from multiple masters, in an attempt to fully utilize available throughput of the attached memories.

A high-performance crossbar switch exists within the device to provide prioritized requests from the different requesters to the EMIFB. If a request is submitted from two or more sources simultaneously, the crossbar switch will forward the highest priority request to the EMIFB first. Upon completion of a request, the crossbar switch again evaluates the pending requests and forwards the highest priority pending request to the EMIFB.

When forwarding read and write commands to the EMIFB, the crossbar uses a priority arbitration scheme. When the EMIFB receives a request, it may or may not be immediately processed due to prioritization of pending refresh cycles. In some cases, the EMIFB will perform one or more auto refresh cycles before processing the request. For details on the EMIFB's internal arbitration between performing requests and performing auto refresh cycles, see Section 2.6.6. For further details regarding master prioritization within the EMIFB command FIFO, see Section 2.6.13.

2.3 Pin Descriptions

Table 1 describes the function of each EMIFB pin.

Table 1. EMIF Pins Used to Access SDRAM

Pins(s)	I/O	Description
EMB_D[x:0]	I/O	**EMIFB data bus.** The number of available data bus pins varies among devices. See your device-specific data manual for details.
EMB_A[x:0]	O	**EMIFB address bus.** When interfacing to an SDRAM device, these pins are primarily used to provide the row and column address to the SDRAM. The number of available address pins depends upon pin multiplexing configuration. See your device-specific data manual for details. The mapping from the internal program address to the external values placed on these pins can be found in Table 15 and Table 16
EMB_BA[1:0]	O	**EMIFB bank address.** When interfacing to an SDRAM device, these pins are used to provide the bank address inputs to the SDRAM. The mapping from the internal program address to the external values placed on these pins can be found in Table 15 and Table 16.
$\overline{\text{EMB\_WE\_DQM[x:0]}}$	O	**Byte enables.** When interfacing to SDRAM, these pins are connected to the DQM pins of the SDRAM to individually enable/disable each of the bytes in a data access.
$\overline{\text{EMB\_WE}}$	O	**Active-low write enable.** When interfacing to SDRAM, this pin is connected to the $\overline{\text{WE}}$ pin of the SDRAM and is used to send commands to the device.
$\overline{\text{EMB\_CS}}$	O	**Active-low chip enable pin for SDRAM devices.** This pin is connected to the chip-select pin of the attached SDRAM device and is used for enabling/disabling commands. By default, the EMIF keeps SDRAM chip select active, even if the EMIF interface is currently idle.

Table 1. EMIF Pins Used to Access SDRAM (continued)

Pins(s)	I/O	Description
EMB_RAS	O	**Active-low row address strobe pin.** This pin is connected to the RAS pin of the attached SDRAM device and is used for sending commands to the device.
EMB_CAS	O	**Active-low column address strobe pin.** This pin is connected to the CAS pin of the attached SDRAM device and is used for sending commands to the device.
EMB_SDCKE	O	**Clock enable pin.** This pin is connected to the CKE pin of the attached SDRAM device and is used for issuing the SELF REFRESH command which places the device in self-refresh mode. See Section 2.6.7 for details.
EMB_CLK	O	**SDRAM clock pin.** This pin is connected to the CLK pin of the attached SDRAM device. See Section 2.1 for details on the clock signal.

2.4 Pin Multiplexing

Refer to device-specific data manual for pin multiplexing details.

2.5 Memory Map

See your device-specific data manual for information describing the device memory-map.

2.6 SDRAM Controller and Interface

The EMIFB can gluelessly interface to most standard SDR SDRAM devices and support such features as self-refresh mode and prioritized refresh. In addition, it provides flexibility through programmable parameters such as the refresh rate, CAS latency, and many SDRAM timing parameters. The following sections include details on how to interface and properly configure the EMIFB to perform read and write operations to externally connected SDR SDRAM devices.

2.6.1 SDRAM Commands

The EMIFB supports the SDRAM commands described in Table 2. Table 3 shows the truth table for the SDRAM commands, and an example timing waveform of the PRE command is shown in Figure 2. EMB_A[10] is pulled low in this example to deactivate only the bank specified by the EMB_BA pins.

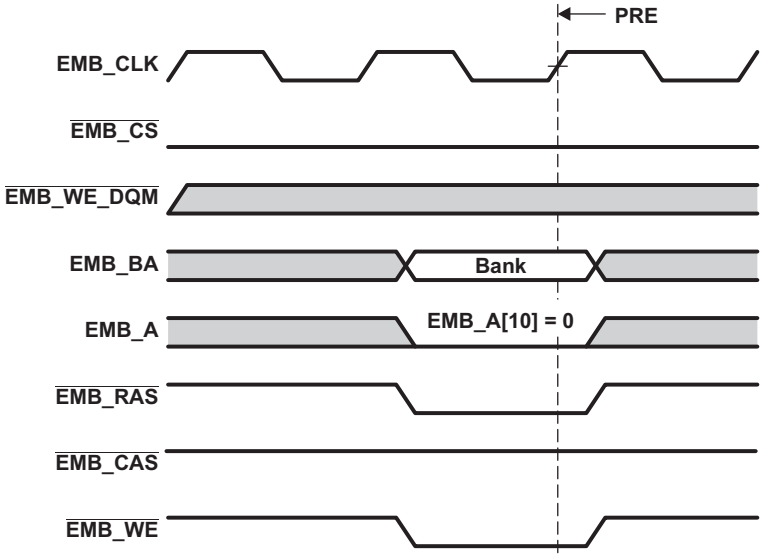
Table 2. EMIF SDRAM Commands

Command	Function
PRE	**Precharge.** Depending on the value of EMB_A[10], the PRE command either deactivates the open row in all banks (EMB_A[10] = 1) or only the bank specified by the EMB_BA[1:0] pins (EMB_A[10] = 0).
ACTV	**Activate**. The ACTV command activates the selected row in a particular bank for the current access.
READ	**Read.** The READ command outputs the starting column address and signals the SDRAM to begin the burst read operation. Address EMB_A[10] is always pulled low to avoid auto precharge. This allows for better bank interleaving performance.
WRT	**Write.** The WRT command outputs the starting column address and signals the SDRAM to begin the burst write operation. Address EMB_A[10] is always pulled low to avoid auto precharge. This allows for better bank interleaving performance.
BT	**Burst terminate**. The BT command is used to truncate the current read or write burst request.
LMR	**Load mode register**. The LMR command sets the mode register of the attached SDRAM devices and is only issued during the SDRAM initialization sequence described in Section 2.6.4.
REFR	**Auto refresh**. The REFR command signals the SDRAM to perform an auto refresh according to its internal address.
SLFR	**Self refresh**. The self refresh command places the SDRAM into self-refresh mode, during which it provides its own clock signal and auto refresh cycles.
NOP	**No operation**. The NOP command is issued during all cycles in which one of the above commands is not issued.

Table 3. Truth Table for SDRAM Commands

SDRAM Pins:	CKE	$\overline{CS}$	$\overline{RAS}$	$\overline{CAS}$	$\overline{WE}$	BA[1:0]	A[12:11]	A[10]	A[9:0]
EMIFB Pins:	EMB_SDCKE	$\overline{EMB\_CS}$	$\overline{EMB\_RAS}$	$\overline{EMB\_CAS}$	$\overline{EMB\_WE}$	EMB_BA[1:0]	EMB_A[12:11]	EMB_A[10]	EMB_A[9:0]
PRE	H	L	L	H	L	Bank/X	X	L/H	X
ACTV	H	L	L	H	H	Bank	Row	Row	Row
READ	H	L	H	L	H	Bank	Column	L	Column
WRT	H	L	H	L	L	Bank	Column	L	Column
BT	H	L	H	H	L	X	X	X	X
LMR	H	L	L	L	L	X	Mode	Mode	Mode
REFR	H	L	L	L	H	X	X	X	X
SLFR	L	L	L	L	H	X	X	X	X
NOP	H	L	H	H	H	X	X	X	X

Figure 2. Timing Waveform of SDRAM PRE Command



2.6.2 Interfacing to SDRAM

The EMIFB supports a glueless interface to SDRAM devices with the following characteristics:

- Pre-charge bit is A[10]
- The number of column address bits is 8, 9, 10 or 11
- The number of row address bits is 13(in case of mobile SDR, number of row address bits can be 9, 10, 11, 12, or 13)
- The number of internal banks is 1, 2 or 4

Figure 3 shows an interface between the EMIFB and a 2M × 16 × 4 bank SDRAM device. In addition, Figure 4 shows an interface between the EMIFB and a 2M × 32 × 4 bank SDRAM device and Figure 5 shows an interface between the EMIFB and two 4M × 16 × 4 bank SDRAM devices. Refer to Table 4, as an example that shows additional list of commonly-supported SDRAM devices and the required connections for the address pins. Note that in Table 4, page size/column size (not indicated in the table) is varied to get the required addressability range.

Figure 3. EMIFB to 2M × 16 × 4 bank SDRAM Interface

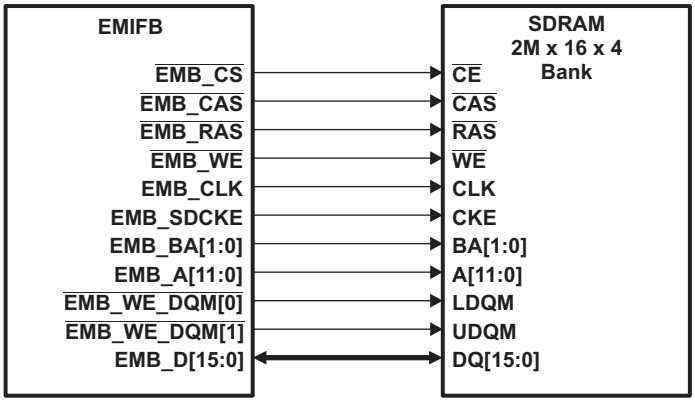


Figure 4. EMIFB to 2M × 32 × 4 bank SDRAM Interface

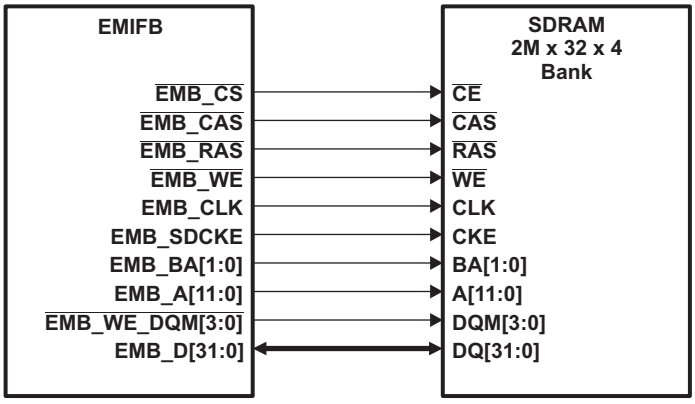


Figure 5. EMIFB to Dual 4M × 16 × 4 bank SDRAM Interface

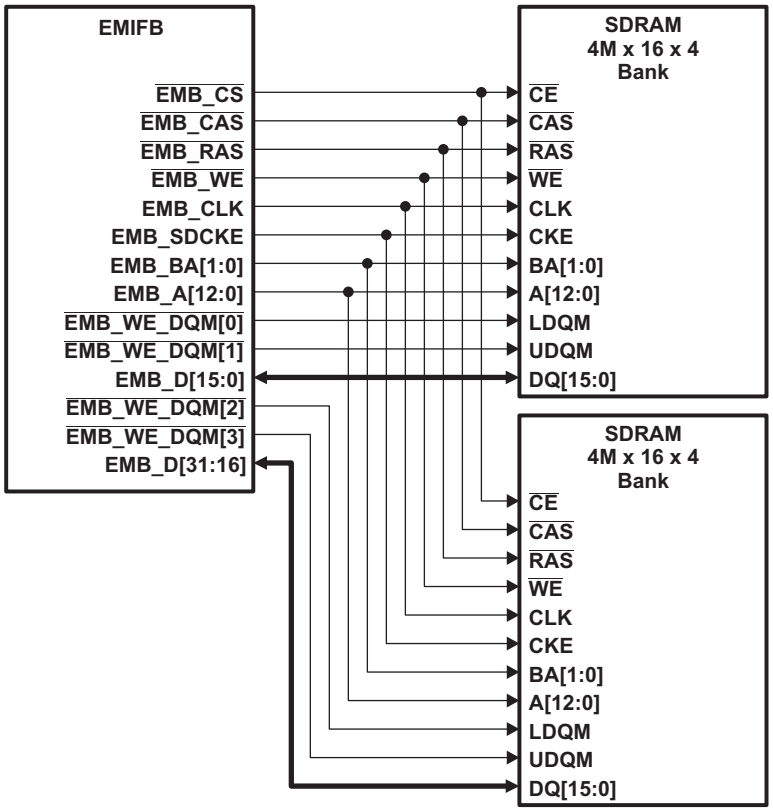


Table 4. Example of 32-bit EMIFB Address Pin Connections

SDRAM Size	Width	Banks		Address Pins
64M bits	×16	4	SDRAM	A[11:0]
			EMIFB	EMB_A[11:0]
	×32	4	SDRAM	A[10:0]
			EMIFB	EMB_A[10:0]
128M bits	×16	4	SDRAM	A[11:0]
			EMIFB	EMB_A[11:0]
	×32	4	SDRAM	A[11:0]
			EMIFB	EMB_A[11:0]
256M bits	×16	4	SDRAM	A[12:0]
			EMIFB	EMB_A[12:0]
	×32	4	SDRAM	A[11:0]
			EMIFB	EMB_A[11:0]
512M bits	×16	4	SDRAM	A[12:0]
			EMIFB	EMB_A[12:0]
	×32	4	SDRAM	A[12:0]
			EMIFB	EMB_A[12:0]

Table 5. Example of 16-bit EMIFB Address Pin Connections

SDRAM Size	Width	Banks		Address Pins
64M bits	×16	4	SDRAM	A[11:0]
			EMIFB	EMB_A[11:0]
128M bits	×16	4	SDRAM	A[11:0]
			EMIFB	EMB_A[11:0]

2.6.3 SDRAM Configuration Registers

The operation of the EMIFB SDRAM interface is controlled by programming the appropriate configuration registers. This section describes the purpose and function of each configuration register, refer to Section 3 for a more detailed description of each register. The following tables list the SDRAM configuration registers, along with a description of each of their programmable fields.

Note: Writing to any of the fields in SDCFG and SDCFG2 causes the EMIFB to abandon whatever it is currently doing and trigger the SDRAM initialization procedure described in Section 2.6.4.

Table 6. Description of the SDRAM Configuration Register (SDCFG)

Parameter	Description
IBANK_POS	**Internal bank position.** Set to 1 to assign internal bank address bits from logical address as shown in Table 17(this addressing scheme is normally used in case of mobile SDRAM). Clear to 0 to assign internal bank address bits from logical address as shown in Table 15 and Table 16 (these addressing schemes are normally used in case of SDR SDRAM) . This bit is writeable only when the BOOT_UNLOCK bit is unlocked.
MSDRAM_ENABLE	**mobile SDR enable.**Both SDREN and MSDRAM_ENABLE should be set to 1 to enable mobile SDR. This bit is writeable only when the BOOT_UNLOCK bit is unlocked.
BOOT_UNLOCK	**Boot Unlock.**Set to 1 to change the values of the fields that are affected by the BOOT_UNLOCK bit.
SDREN	**SDR Enable.** This bit enables EMIFB to interface to SDRAM type memories. This bit is set to 1 by default.
TIMUNLOCK	**Timing Unlock.**Controls the write permission settings for the SDRAM timing 1 register (SDTIM1) and SDRAM timing 2 register (SDTIM2)
NM	**Narrow Mode.** This bit defines the width of the data bus between the EMIF and the attached SDRAM device. When set to 1, the data bus is set to 16-bits; when cleared to 0, the data bus is set to 32-bits.
CL	**CAS latency.** This field defines the number of clock cycles between when an SDRAM issues a READ command and when the first piece of data appears on the bus. The value in this field is sent to the attached SDRAM device via the LOAD MODE REGISTER command during the SDRAM initialization procedure as described in Section 2.6.4. Only values of 2h (CAS latency = 2) and 3h (CAS latency = 3) are supported and are written to this field. A 1 must be simultaneously written to the TIMUNLOCK bit field of SDCFG in order to write to the CL bit field.
EBANK	**Number of External SDRAM Banks (or chip selects).** This field defines the number of chip selects are utilized on the SDRAM interface: • When EBANK = 0, CS[0] is used (single external bank). Always write 0 to this field.
IBANK	**Number of Internal SDRAM Banks.** This field defines the number of banks inside the attached SDRAM devices in the following way: • When IBANK = 0, 1 internal bank is used • When IBANK = 1h, 2 internal banks are used • When IBANK = 2h, 4 internal banks are used This field value affects the mapping of logical addresses to SDRAM row, column, and bank addresses. See Section 2.6.12 for details.

Table 6. Description of the SDRAM Configuration Register (SDCFG) (continued)

Parameter	Description
PAGESIZE	**Page Size.** This field defines the internal page size of the attached SDRAM devices in the following way: • When PAGESIZE = 0, 256-word pages are used, requiring 8 column address bits. • When PAGESIZE = 1h, 512-word pages are used, requiring 9 column address bits. • When PAGESIZE = 2h, 1024-word pages are used, requiring 10 column address bits. • When PAGESIZE = 3h, 2048-word pages are used, requiring 11 column address bits. This field value affects the mapping of logical addresses to SDRAM row, column, and bank addresses. See Section 2.6.12 for details.

Table 7. Description of the SDRAM Refresh Control Register (SDRFC)

Parameter	Description
LP_MODE	**Low Power Mode**. This bit enables the self-refresh mode of the attached SDRAM devices (which is the lowest power mode).
MCLKSTOP_EN	**mclk stop enable.** mclk can stopped only if this bit is set.
SR_PD	**Self Refresh/ Power Down select.** This bit along with LP_MODE determines if SDRAM is to be placed in self-refresh/power-down mode.
REFRESH_RATE	**Refresh Rate**. This field controls the rate at which attached SDRAM devices will be refreshed. The following equation can be used to determine the required value of REFRESH_RATE for an SDRAM device: • REFRESH_RATE = (EMIFB clock rate)/(Required SDRAM Refresh Rate) More information about the operation of the SDRAM refresh controller can be found in Section 2.6.6.

Table 8. Description of the SDRAM Timing 1 Register (SDTIM1)

Parameter	Description
T_RFC T_RP T_RCD T_WR T_RAS T_RC T_RRD	**SDRAM Timing Parameters.** These fields configure the EMIFB to comply with the AC timing requirements of the attached SDRAM devices. This allows the EMIFB to avoid violating SDRAM timing constraints and to more efficiently schedule its operations. More details about each of these parameters can be found in the register description in Section 3.4. These parameters are set to satisfy the corresponding timing requirements found in the SDRAM's datasheet.

Table 9. Description of the SDRAM Timing 2 Register (SDTIM2)

Parameter	Description
T_RAS_MAX	Maximum number of refresh_rate intervals from Activate to Precharge command.
T_XS	**Self Refresh Exit Parameter.** The T_XS field of this register informs the EMIFB about the minimum number of EMB_CLK cycles required between exiting Self Refresh and issuing any command. This parameter is set to satisfy the t_{XSR} value for the attached SDRAM device.
T_CKE	The T_CKE field fixes the minimum time between CKE transitions. This parameter is set to satisfy the t_{RAS} value for the attached SDRAM device.

Table 10. Description of the SDRAM Configuration 2 Register (SDCFG2)

Parameter	Description
PASR	**Partial Array Self Refresh.** These bits get loaded into the Extended Mode Register of a mobile SDRAM during initialization. A write to this field will cause the EMIFB to start the SDRAM initialization sequence.
ROWSIZE	**Row Size.** Defines the number of row address bits of connected SDRAM devices. This bit is used only in case of mobile SDRAM. A write to this field will cause the EMIFB to start the SDRAM initialization sequence.

2.6.4 SDRAM/mobile SDRAM Auto-Initialization Sequence

The EMIFB automatically performs an SDRAM initialization sequence, regardless of whether it is interfaced to an SDRAM device, when the following event occurs:

- A write is performed to any of the two least significant bytes of the SDRAM configuration register (SDCFG)
- In case of mobile SDR, initialization sequence also starts when a write is performed to SDRAM configuration 2 register (SDCFG2)

An SDRAM/mobile SDR initialization sequence consists of the following steps:

1. First, software must set the SDREN bit (in case of mobile SDRAM, both SDREN and MSDRAM_ENABLE should be set to 1) in the SDRAM configuration register (SDCFG) (assuming clocking and pin multiplexing are already configured accordingly).
2. If the initialization sequence is activated by a write to SDCFG, and if any of the SDRAM banks are open, the EMIFB issues a PRE command with EMB_A[10] held high to indicate all banks. This is done so that the maximum ACTV to PRE timing for an SDRAM is not violated.
3. The EMIFB drives EMB_SDCKE high and begins continuously issuing NOP commands until eight SDRAM refresh intervals have elapsed. An SDRAM refresh interval is equal to the value of the REFRESH_RATE field of the SDRAM refresh control register (SDRFC), divided by the frequency of EMB_CLK (REFRESH_RATE/f_{CLK}). This step is used to avoid violating the Power-up constraint of most SDRAM devices that requires 200 µs (sometimes 100 µs) between receiving stable Vdd and CLK and the issuing of a PRE command. Depending on the frequency of EMB_CLK, this step may or may not be sufficient to avoid violating the SDRAM constraint. See Section 2.6.5 for more information.
4. After the refresh intervals have elapsed, the EMIFB issues a PRE command with EMB_A[10] held high to indicate all banks.
5. The EMIFB issues eight AUTO REFRESH commands.
6. If initialization sequence is of mobile SDRAM, EMIFB issues LMR command with EMB_A[6:0] pins set as described in Table 11.
7. Then, EMIFB issues the LMR command with the EMB_A[9:0] pins set as described in Table 12. This step is executed for both SDRAM/mobile SDRAM.
8. Finally, the EMIFB performs an auto refresh cycle, which consists of the following steps:
   a. Issuing a PRE command with EMB_A[10] held high if any banks are open
   b. Issuing a REF command
   c. Interface is idle (awaiting access)

Table 11. mobile SDRAM LOAD MODE REGISTER Command

A[6:5]	A[4:3]	A[2:0]
0 (SDRAM drive strength; 0= full drive strength)	0 (Internal Temperature Compensated Self Refresh)	These bits are set according to the PASR field in the SDRAM configuration 2 register (SDCFG2).

Table 12. SDRAM/mobile SDRAM LOAD MODE REGISTER Command

A[9:7]	A[6:4]	A[3]	A[2:0]
0 (Write bursts are of the programmed burst length in EMB_A[2:0])	These bits control the CAS latency of the SDRAM and are set according to CL field in the SDRAM configuration register (SDCFG) as follows: • If CL = 2h, EMB_A[6:4] = 2h (CAS latency = 2) • If CL = 3h, EMB_A[6:4] = 3h (CAS latency = 3)	0 (Sequential Burst Type. Interleaved Burst Type not supported)	These bits control the burst length of the SDRAM and are set according to the NM field in the SDRAM configuration register (SDCFG) as follows: • If NM = 0, EMB_A[2:0] = 2h (Burst Length = 4) • If NM = 1, EMB_A[2:0] = 3h (Burst Length = 8)

2.6.5 SDRAM Configuration Procedure

After initial power-on, follow the procedure listed below before performing any EMIFB memory requests. Note that the SDRAM power-up constraint specifies that 200 µs must exist between receiving stable Vdd and CLK and the issuing of a PRE command. Initialization software and system design must ensure that this constraint is met before executing the initialization procedure.

1. Place the SDRAM into Self-Refresh Mode by setting the LP_MODE bit and SR_PD bit of the SDRAM refresh control register (SDRFC) to 1 and 0, respectively. Place the SDRAM into Self-Refresh mode when changing the frequency of EMB_CLK to avoid incurring the 200 µs power-up constraint again.
2. Program the PLL controller and configure the EMIFB clock mux selection (in the System Configuration Module) to attain the desired EMB_CLK clock frequency. Refer to the device data manual for details on programming the PLL controller. The frequency of the memory clock must meet the timing requirements in the SDRAM manufacturer's documentation and the timing limitations shown in the electrical specifications of the device data manual.
3. Enable SDR mode of the EMIFB by writing 1 to the SDREN bit (write 1 to both SDREN and MSDRAM_ENABLE to enable mobile SDR) in the SDRAM configuration register (SDCFG). Also ensure that pin multiplexing is properly configured.
4. Program SDTIM1 and SDTIM2 to satisfy the timing requirements for the attached SDRAM device. Take the timing parameters from the SDRAM datasheet.
5. Program the REFRESH_RATE field of SDRFC to match that of the attached device's refresh interval. See Section 2.6.6.1 for details on determining the appropriate value.
6. Program SDCFG to match the characteristics of the attached SDRAM device. This will cause the auto-initialization sequence in Section 2.6.4 to be re-run. This second initialization generally takes much less time due to the increased frequency of EMB_CLK.

After following the above procedure, the EMIFB is ready to perform accesses to the attached SDRAM device. If a frequency change is desired after this configuration has been executed, first put the SDRAM into Self-Refresh mode using a byte-write to the upper byte of SDCFG to avoid restarting the SDRAM auto-initialization sequence. Then release the SDRAM from self-refresh mode and repeat steps 4 through 6 of the above procedure.

2.6.6 EMIFB Refresh Controller

An SDRAM device requires that each of its rows be refreshed at a minimum required rate. The EMIFB can meet this constraint by performing auto refresh cycles at or above this required rate. An auto refresh cycle consists of issuing a PRE command to all banks of the SDRAM device followed by issuing a REFR command. To inform the EMIFB of the required rate for performing auto refresh cycles, the REFRESH_RATE field of the SDRAM refresh control register (SDRFC) must be programmed. The EMIF will use this value along with two internal counters to automatically perform auto refresh cycles at the required rate. The auto refresh cycles cannot be disabled, even if the EMIFB is not interfaced with an SDRAM. The remainder of this section details the EMIFB's refresh scheme and provides an example for determining the appropriate value to place in the REFRESH_RATE field of SDRFC.

The two counters used to perform auto-refresh cycles are a 13-bit refresh interval counter and a 4-bit refresh backlog counter. After SDREN = 1 and upon writing to the REFRESH_RATE field, the refresh interval counter is loaded with the value from REFRESH_RATE field and begins decrementing, by one, each EMIFB clock cycle. When the refresh interval counter reaches zero, the following actions occur:

- The refresh interval counter is reloaded with the value from the REFRESH_RATE field and restarts decrementing.
- The 4-bit refresh backlog counter increments unless it has already reached its maximum value.

The refresh backlog counter records the number of auto refresh cycles that the EMIFB currently has outstanding. This counter is decremented by one each time an auto refresh cycle is performed and incremented by one each time the refresh interval counter expires. The refresh backlog counter saturates at the values of 0000b and 1111b. The EMIFB uses the refresh backlog counter to determine the urgency with which an auto refresh cycle is to be performed. The four levels of urgency are described in Table 13. This refresh scheme allows the required refreshes to be performed with minimal impact on access requests.

Table 13. Refresh Urgency Levels

Urgency Level	Refresh Backlog Counter Range	Action Taken
Refresh May	1-3	An auto-refresh cycle is performed only if the EMIFB has no requests pending and none of the SDRAM banks are open.
Refresh Release	4-7	An auto-refresh cycle is performed if the EMIFB has no requests pending, regardless of whether any SDRAM banks are open.
Refresh Need	8-11	An auto-refresh cycle is performed at the completion of the current access unless there are read requests pending.
Refresh Must	12-15	Multiple auto-refresh cycles are performed at the completion of the current access until the Refresh Release urgency level is reached. At that point, the EMIFB can begin servicing any new read or write requests.

2.6.6.1 Determining the Appropriate Value for the REFRESH_RATE Field

The value programmed into the REFRESH_RATE field of SDRFC can be calculated by using the frequency of the EMB_CLK signal (f_{CLK}) and the required refresh rate of the SDRAM ($f_{Refresh}$). The following formula can be used:

$REFRESH\_RATE \leq f_{CLK} / f_{Refresh}$

The SDRAM datasheet often communicates the required SDRAM Refresh Rate in terms of the number of REFR commands required in a given time interval. The required SDRAM Refresh Rate in the formula above can be therefore be calculated by dividing the number of required cycles per time interval (n_{cycles}) by the time interval given in the datasheet ($t_{Refresh\ Period}$) :

$f_{Refresh} = n_{cycles} / t_{Refresh\ Period}$

Combining these formulas, the value programmed into the REFRESH_RATE field can be computed as:

$REFRESH\_RATE \leq f_{CLK} \times t_{Refresh\ Period} / n_{cycles}$

The following example illustrates calculating the value of REFRESH_RATE. Given that:

- f_{CLK} = 133 MHZ (frequency of the EMIFB clock)
- $t_{Refresh\ Period}$ = 64 ms (required refresh interval of the SDRAM)
- n_{cycles} = 8192 (number of cycles in a refresh interval for the SDRAM)

REFRESH_RATE can be calculated as:

$REFRESH\_RATE = 133\ MHZ \times 64\ ms/8192$

$REFRESH\_RATE = 1039.06$

REFRESH_RATE = 1039 cycles = 40Fh cycles

2.6.7 Self-Refresh Mode

The EMIFB can be programmed to enter the self-refresh state by setting the LP_MODE bit and SR_PD bit of the SDRAM refresh control register (SDRFC) to 1 and 0, respectively. This will cause the EMIFB to issue the SLFR command after completing any outstanding SDRAM access requests and clearing the refresh backlog counter by performing one or more auto refresh cycles. This places the attached SDRAM device into self-refresh mode in which it consumes a minimal amount of power while performing its own refresh cycles.

While in the self-refresh state, the EMIFB continues to service register accesses as normal.

The EMIFB will exit from the self-refresh state, if any of the following events occur:

- The LP_MODE bit of SDRFC is cleared to 0
- The SR_PD bit is set to 1
- An SDRAM accesses is requested

The EMIFB exits from the self-refresh state by driving EMB_SDCKE high and performing an auto refresh cycle.

The attached SDRAM device must be placed into self-refresh mode when changing the frequency of EMB_CLK using the PLL Controller. If the frequency of EMB_CLK changes while the SDRAM is not in self-refresh mode, the memory must be reinitialized.

During Self- refresh, if memory/register access request is made, EMIFB comes out of self-refresh state (driving EMB_SDCKE high) and executes the requests; after which it again goes back to self-refresh state (driving EMB_SDCKE low).

To use Partial Array Self Refresh for mobile SDR, PASR bits in the SDRAM configuration 2 register must be appropriately programmed. The EMIFB performs bank interleaving. Since the SDRAM is partially refreshed during Partial Array Self Refresh, it is the responsibility of software to move critical data into the banks that are going to be refreshed during Partial Array Self Refresh.

2.6.8 Power-Down Mode

To support low-power modes, the EMIFB can be requested to issue a POWERDOWN command to the SDRAM by setting both the LP_MODE and SR_PD bits in the SDRAM refresh control register (SDRFC) to 1. When this bit is set, the EMIFB will continue normal operation until all outstanding memory access requests have been serviced and the SDRAM refresh backlog (if there is one) has been cleared. At this point the EMIFB will enter the power-down state. Upon entering this state the EMIF will issue a POWERDOWN command (same as a NOP command but driving EMB_SDCKE low on the same cycle). The EMIFB then maintains EMB_SDCKE low until it exits the power-down state.

During the power-down state, the EMIFB services synchronous memory and register accesses as normal.

The EMIFB will exit from the power-down state, if any of the following events occur:

- The LP_MODE bit of SDRFC is cleared to 0
- The SR_PD bit is cleared to 0
- An SDRAM accesses is requested
- Refresh (REFR) command is to be sent to SDRAM.

During power-down, if memory/register access request is made, EMIFB comes out of the power-down state (driving EMB_SDCKE high) and executes the requests; after which it again goes back to the power-down state (driving EMB_SDCKE low).

2.6.9 Partial Array Self Refresh for mobile SDRAM

This is applicable only to mobile SDRAM, when using the addressing scheme as described in Table 17. For additional power savings during self-refresh, the partial array self-refresh (PASR) feature of mobile SDR allows to select the amount of memory that will be refreshed during self-refresh. Use the partial array self-refresh (PASR) bit field in the SDRAM configuration 2 register (SDCFG2) to select the amount of memory to refresh during self-refresh. As shown in Table 14 you may select either 4, 2, 1, 1/2, or 1/4 bank(s). The PASR bits are loaded into the extended mode register of the mobile SDR device, during autoinitialization (see Section 2.6.4). The EMIFB performs bank interleaving when the internal bank position (IBANKPOS) bit in SDRAM configuration register (SDCFG) is cleared to 0. Since the SDRAM banks are only partially refreshed during partial array self-refresh, it is recommended that you set IBANKPOS to 1 to avoid bank interleaving. Refer to Section 2.6.12 for more information on IBANKPOS and addressing mapping in general.

Table 14. PASR Bitfield in SDRAM Configuration 2 Register (SDCFG2) Configuration

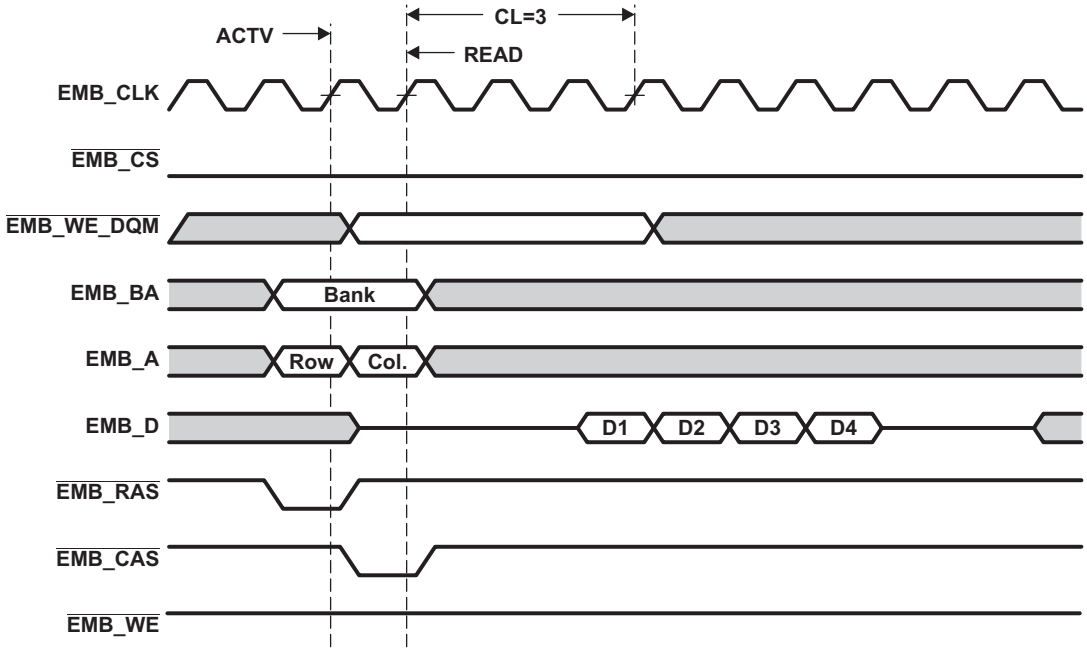
Bit Field	Bit Value	Bit Description
PASR	0	Refresh banks 0, 1, 2, and 3
	1h	Refresh banks 0 and 1
	2h	Refresh bank 0
	3h	Reserved
	4h	Reserved
	5h	Refresh 1/2 of bank 0
	6h	Refresh 1/4 of bank 0
	7h	Reserved

2.6.10 SDRAM Read Operation

When the EMIFB receives a read request to SDRAM , it performs one or more read access cycles. A read access cycle begins with the issuing of the ACTV command to select the desired bank and row of the SDRAM device. After the row has been opened, the EMIFB proceeds to issue a READ command while specifying the desired bank and column address. EMB_A[10] is held low during the READ command to avoid auto-precharging. The READ command signals the SDRAM device to start bursting data from the specified address while the EMIFB issues NOP commands. Following a READ command, the CL field of the SDRAM configuration register (SDCFG) defines how many delay cycles will be present before the read data appears on the data bus. This is referred to as the CAS latency.

Figure 6 shows the signal waveforms for a basic SDRAM read operation in which a burst of data is read from a single page. When the EMIFB SDRAM interface is configured to 32-bit by clearing the NM bit of the SDRAM configuration register (SDCFG) to 0, a burst size of four is used. When configured to 16-bit by setting NM to 1, a burst size of eight is used. Figure 6 shows a burst size of four.

Figure 6. Timing Waveform for Basic SDRAM Read Operation



The EMIFB will truncate a series of bursting data if the remaining addresses of the burst are not required to complete the request. The EMIFB can truncate the burst in three ways:

- By issuing another READ to the same page in the same bank.
- By issuing a PRE command in order to prepare for accessing a different page of the same bank.
- By issuing a BT command in order to prepare for accessing a page in a different bank.

Several other pins are also active during a read access. The $\overline{\text{EMB\_WE\_DQM}}$[3:0] pins are driven low during the READ commands and are kept low during the NOP commands that correspond to the burst request. The state of the other EMIFB pins during each command can be found in Table 3.

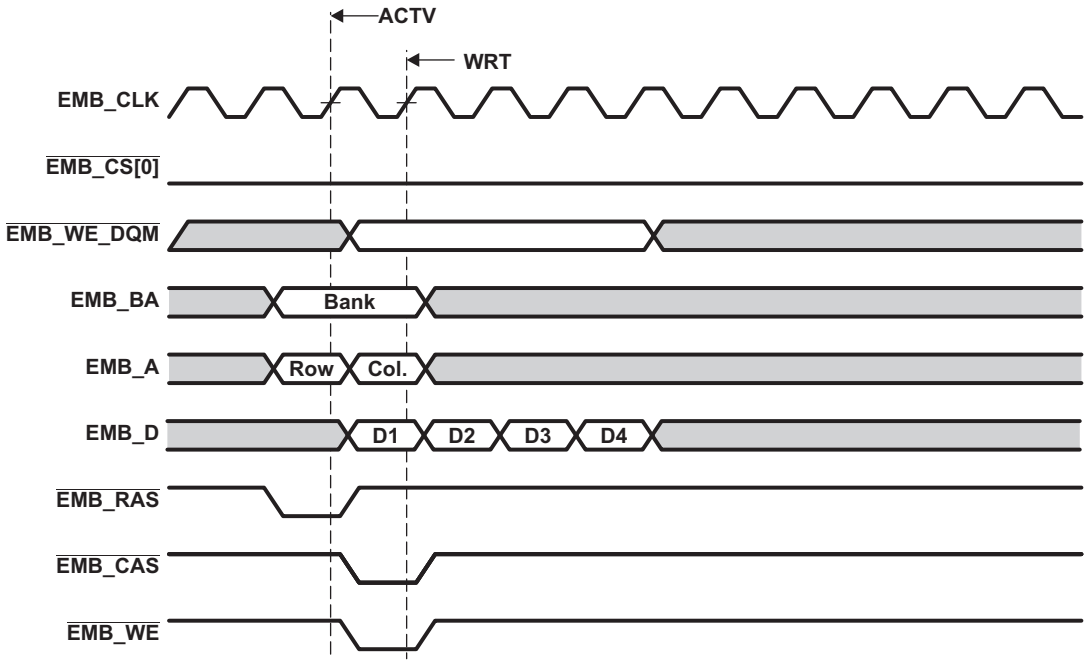
The EMIFB schedules its commands based on the timing information that is provided to it in the SDRAM timing registers (SDTIM1 and SDTIM2). The values for the timing parameters in this register are chosen to satisfy the timing requirements listed in the SDRAM datasheet. The EMIFB uses this timing information to avoid violating any timing constraints related to issuing commands. This is commonly accomplished by inserting NOP commands between various commands during an access. Refer to the register description of SDTIM1 and SDTIM2 for more details on the various timing parameters.

2.6.11 SDRAM Write Operations

When the EMIFB receives a write request to SDRAM , it performs one or more write-access cycles. A write-access cycle begins with the issuing of the ACTV command to select the desired bank and row of the SDRAM device. After the row has been opened, the EMIFB proceeds to issue a WRT command while specifying the desired bank and column address. EMB_A[10] is held low during the WRT command to avoid auto-precharging. The WRT command signals the SDRAM device to start writing a burst of data to the specified address while the EMIFB issues NOP commands. The associated write data will be placed on the data bus in the cycle concurrent with the WRT command and with subsequent burst continuation NOP commands.

Figure 7 shows the signal waveforms for a basic SDRAM write operation in which a burst of data is read from a single page. When the EMIFB SDRAM interface is configured to 32-bit by clearing the NM bit of the SDRAM configuration register (SDCFG) to 0, a burst size of four is used. When configured to 16-bit by setting NM to 1, a burst size of eight is used. Figure 7 shows a burst size of four.

Figure 7. Timing Waveform for Basic SDRAM Write Operation



The EMIFB will truncate a series of bursting data if the remaining addresses of the burst are not part of the write request. The EMIFB can truncate the burst in three ways:

- By issuing another WRT to the same page
- By issuing a PRE command in order to prepare for accessing a different page of the same bank
- By issuing a BT command in order to prepare for accessing a page in a different bank

Several other pins are also active during a write access. The EMB_WE_DQM[3:0] pins are driven to select which bytes of the data word will be written to the SDRAM device. They are also used to mask out entire undesired data words during a burst access. The state of the other EMIFB pins during each command can be found in Table 3.

EMIFB schedules its commands based on the timing information that is provided to it in the SDRAM timing registers (SDTIM1 and SDTIM2). The values for the timing parameters in this register are chosen to satisfy the timing requirements listed in the SDRAM datasheet. EMIFB uses this timing information to avoid violating any timing constraints related to issuing commands. This is commonly accomplished by inserting NOP commands during various cycles of an access. Refer to the register description of SDTIM1 and SDTIM2 for more details on the various timing parameters.

2.6.12 Mapping from Logical Address to EMIFB Pins

When the EMIFB receives an SDRAM access request, it must convert the address of the access into the appropriate signals to send to the SDRAM device. The details of an example address mapping are shown in Table 15 for 32-bit operation and in Table 16 for 16-bit operation. (In both the examples, a 13-bit row address is used to calculate the maximum reach. See your device-specific data manual to know the possible values of IBANK and PAGESIZE for EMIFB). Using the settings of the IBANK and PAGESIZE fields of the SDRAM configuration register (SDCFG), the EMIFB determines which bits of the logical address will be mapped to the SDRAM row, column, and bank addresses.

As the logical address is incremented by one word (32-bit operation) or one halfword (16-bit operation), the column address is likewise incremented by one until a page boundary is reached. When the logical address increments across a page boundary, the EMIFB moves into the same page in the next bank of the attached device by incrementing the bank address EMB_BA and resetting the column address. The page in the previous bank is left open until it is necessary to close it. This method of traversal through the SDRAM banks helps maximize the number of open banks inside of the SDRAM and results in an efficient use of the device. There is no limitation on the number of banks than can be open at one time, but only one page within a bank can be open at a time. To use such an addressing scheme, clear the internal bank position (IBANK_POS) bit in SDCFG to 0. This addressing scheme is used when EMIFB memory controller is configured to interface with SDR SDRAM.

The EMIFB uses the $\overline{\text{EMB\_WE\_DQM}}$ pins during a WRT command to mask out selected bytes or entire words. The $\overline{\text{EMB\_WE\_DQM}}$ pins are always low during a READ command.

When using mobile SDRAM, set IBANK_POS = 1, and this uses an addressing scheme as described in Table 17. See device data manual to know possible values of ROWSIZE, IBANK, and PAGESIZE for EMIFB configured to interface with mobile SDRAM device.

When the IBANK_POS bit is set to 1, the PAGESIZE, ROWSIZE, and IBANK fields control the mapping of the logical source address of the memory controller to the column, row, and bank address bits of the SDRAM device. Table 17 shows which source address bits map to the SDRAM column, row, and bank address bits for all combinations of PAGESIZE, ROWSIZE, and IBANK.

When the IBANK_POS bit is set to 1, the effect of the address-mapping scheme is that as the source address increments across an SDRAM page boundary, the memory controller proceeds to the next page in the same bank. This movement along the same bank continues until all the pages have been accessed in the same bank. The memory controller then proceeds to the next bank in the device. Since, in this address mapping scheme, the memory controller can keep only one bank open, this scheme is lower in performance than the case when IBANK_POS is cleared to 0. Therefore, this case is only recommended to be used with Partial Array Self-refresh for mobile SDR SDRAM where performance may be traded-off for power savings.

Table 15. Example Mapping from Logical Address to EMIFB Pins for 32-bit SDRAM

REACH (MB)	IBANK	PAGE SIZE	31	30	29	28	27	26	25	24	23	22:15	14	13	12	11	10	9:2	1:0
8	0	0	-									Row Address						Column Address	$\overline{\text{WE\_DQM}}$[3:0]
16	1	0	-									Row Address					BA[0]	Column Address	$\overline{\text{WE\_DQM}}$[3:0]
32	2	0	-									Row Address				BA[1:0]		Column Address	$\overline{\text{WE\_DQM}}$[3:0]
16	0	1	-									Row Address					Column Address		$\overline{\text{WE\_DQM}}$[3:0]
32	1	1	-									Row Address				BA[0]	Column Address		$\overline{\text{WE\_DQM}}$[3:0]
64	2	1	-									Row Address			BA[1:0]		Column Address		$\overline{\text{WE\_DQM}}$[3:0]
32	0	2	-									Row Address				Column Address			$\overline{\text{WE\_DQM}}$[3:0]
64	1	2	-									Row Address			BA[0]	Column Address			$\overline{\text{WE\_DQM}}$[3:0]
128	2	2	-									Row Address		BA[1:0]		Column Address			$\overline{\text{WE\_DQM}}$[3:0]
64	0	3	-									Row Address				Column Address			$\overline{\text{WE\_DQM}}$[3:0]
128	1	3	-									Row Address		BA[0]		Column Address			$\overline{\text{WE\_DQM}}$[3:0]
256	2	3	-									Row Address		BA[1:0]		Column Address			$\overline{\text{WE\_DQM}}$[3:0]

Table 16. Example Mapping from Logical Address to EMIFB Pins for 16-bit SDRAM

REACH (MB)	IBANK	PAGE SIZE	31	30	29	28	27	26	25	24	23	22	21:14	13	12	11	10	9	8:1	0
4	0	0	-										Row Address						Column Address	$\overline{\text{WE\_DQM}}$[1:0]
8	1	0	-										Row Address					BA[0]	Column Address	$\overline{\text{WE\_DQM}}$[1:0]
16	2	0	-										Row Address				BA[1:0]		Column Address	$\overline{\text{WE\_DQM}}$[1:0]
8	0	1	-										Row Address					Column Address		$\overline{\text{WE\_DQM}}$[1:0]
16	1	1	-										Row Address				BA[0]	Column Address		$\overline{\text{WE\_DQM}}$[1:0]
32	2	1	-										Row Address			BA[1:0]		Column Address		$\overline{\text{WE\_DQM}}$[1:0]
16	0	2	-										Row Address				Column Address			$\overline{\text{WE\_DQM}}$[1:0]
32	1	2	-										Row Address			BA[0]	Column Address			$\overline{\text{WE\_DQM}}$[1:0]
64	2	2	-										Row Address		BA[1:0]		Column Address			$\overline{\text{WE\_DQM}}$[1:0]
32	0	3	-										Row Address			Column Address				$\overline{\text{WE\_DQM}}$[1:0]
64	1	3	-										Row Address		BA[0]	Column Address				$\overline{\text{WE\_DQM}}$[1:0]
128	2	3	-										Row Address		BA[1:0]	Column Address				$\overline{\text{WE\_DQM}}$[1:0]

Note: The upper bit of the Row Address is used only when addressing 256-Mbit and 512-Mbit SDRAM memories.

Table 17. Example Mapping from Logical Address to EMIFB Pins for mobile SDRAM

31	N = 1 for 16-bit mobile SDRAM N = 2 for 32-bit mobile SDRAM	N

Bank Address	Row Address	Column Address	Data Mask
# of bits defined by IBANK	# of bits defined by ROWSIZE	# of bits defined by PAGESIZE	$\overline{\text{WE\_DQM}}$[x:0]
IBANK = 0 => 0 bit	ROWSIZE = 0 => 9 bits	PAGESIZE = 0 => 8 bits	for N = 1, x = 1
IBANK = 1 => 1 bit	ROWSIZE = 1 => 10 bits	PAGESIZE = 1 => 9 bits	for N = 2, x = 3
IBANK = 2 => 2 bits	ROWSIZE = 2 => 11 bits	PAGESIZE = 2 => 10 bits	
	ROWSIZE = 3 => 12 bits	PAGESIZE = 3 => 11 bits	
	ROWSIZE = 4 => 13 bits		

2.6.13 SDRAM Memory Controller FIFO and Prioritization Considerations

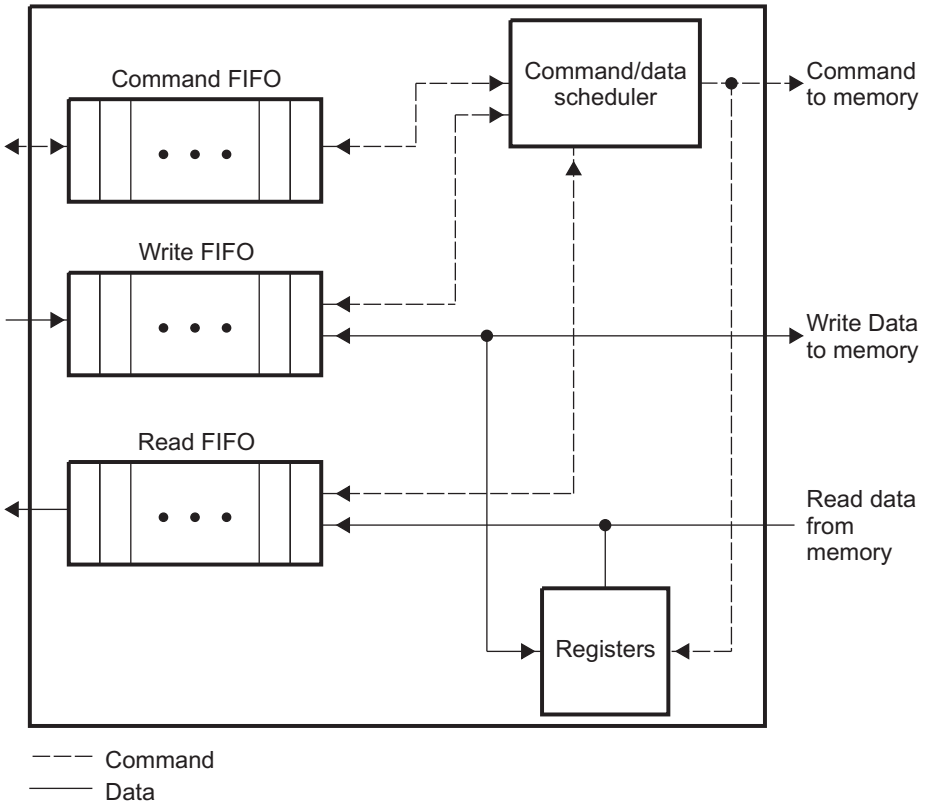
To move data efficiently from on-chip resources to external SDRAM memory, the EMIFB memory controller makes use of a command FIFO, a write FIFO, a read FIFO, and command and data schedulers. Table 18 describes the purpose of each FIFO.

Figure 8 shows the block diagram of the SDRAM memory controller FIFOs. Commands, write data, and read data arrive at the SDRAM memory controller parallel to each other. The same peripheral bus is used to write and read data from external memory as well as internal memory-mapped registers.

Table 18. SDRAM Memory Controller FIFO Description

FIFO	Description	Depth (32-bit words)
Command	Stores all commands coming from on-chip requesters	7
Write	Stores write data coming from on-chip requesters to memory	11
Read	Stores read data coming from memory to on-chip requesters	15

Figure 8. EMIFB Memory Controller FIFO Block Diagram



2.6.13.1 Command Ordering and Scheduling (Advanced Concept)

The SDRAM memory controller performs command re-ordering and scheduling in an attempt to achieve efficient transfers with maximum throughput. The goal is to maximize the utilization of the data, address, and command buses while hiding the overhead of opening and closing EMIFB SDRAM rows. Command re-ordering takes place within the command FIFO.

Typically, a given master issues commands on a single priority. EDMA transfer controller read and write ports are different masters. The SDRAM memory controller first reorders commands from each master based on the following rules:

- Selects the oldest command (first command in the queue)
- Selects a read before a write if:
  - The read is to a different block address (2048 bytes) than the write
  - The read has greater or equal priority

The second bullet above may be viewed as an exception to the first bullet. This means that for an individual master, all of its commands will complete from oldest to newest, with the exception that a read may be advanced ahead of an older, lower or equal priority write. Following this scheduling, each master may have one command ready for execution.

Next, the SDRAM memory controller examines each of the commands selected by the individual masters and performs the following reordering:

- Among all pending reads, selects reads to rows already open. Among all pending writes, selects writes to rows already open.
- Selects the highest priority command from pending reads and writes to open rows. If multiple commands have the highest priority, then the SDRAM memory controller selects the oldest command.

The SDRAM memory controller may now have a final read and write command. If the Read FIFO is not full, then the read command will be performed before the write command, otherwise the write command will be performed first.

Besides commands received from on-chip resources, the SDRAM memory controller also issues refresh commands. The SDRAM memory controller attempts to delay refresh commands as long as possible to maximize performance while meeting the SDRAM refresh requirements. As the SDRAM memory controller issues read, write, and refresh commands to SDRAM memory, it adheres to the following rules:

1. Refresh request resulting from the Refresh Must level of urgency being reached
2. Read request without a higher priority write (selected from above reordering algorithm)
3. Refresh request resulting from the Refresh Need level of urgency being reached
4. Write request (selected from above reordering algorithm)
5. Refresh request resulting from Refresh May level of urgency being reached
6. Request to enter self-refresh mode

The following results from the above scheduling algorithm:

- All writes from a single master will complete in order
- All reads from a single master will complete in order
- From the same master, any read to the same location (or within 2048 bytes) as a previous write will complete in order

2.6.13.2 Command Starvation

The reordering and scheduling rules listed above may lead to command starvation, which is the prevention of certain commands from being processed by the SDRAM memory controller. Command starvation results from the following conditions:

- A continuous stream of high-priority read commands can block a low-priority write command.
- A continuous stream of SDRAM commands to a row in an open bank can block commands to the closed row in the same bank.

To avoid these conditions, the SDRAM memory controller can momentarily raise the priority of the oldest command in the command FIFO after a set number of transfers have been made. The PRIO_RAISE bit field in the peripheral bus burst priority register (BPRIO) sets the number of the transfers that must be made before the SDRAM memory controller will raise the priority of the oldest command.

2.6.13.3 Possible Race Condition

A race condition may exist when certain masters write data to the SDRAM memory controller. For example, if master A passes a software message via a buffer in SDRAM memory and does not wait for indication that the write completes, when master B attempts to read the software message it may read stale data and therefore receive an incorrect message. In order to confirm that a write from master A has landed before a read from master B is performed, master A must wait for the write completion status from the SDRAM memory controller before indicating to master B that the data is ready to be read. If master A does not wait for indication that a write is complete, it must perform the following workaround:

1. Perform the required write.
2. Perform a dummy write to the SDRAM memory controller SDRAM status register.
3. Perform a dummy read to the SDRAM memory controller SDRAM status register.
4. Indicate to master B that the data is ready to be read after completion of the read in step 3. The completion of the read in step 3 ensures that the previous write was done.

The EDMA peripheral does not need to implement the above workaround. If a peripheral is not listed here, then the above workaround is required.

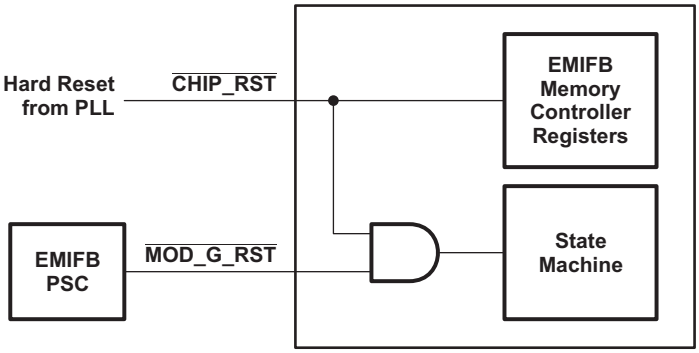
2.7 Reset and Initialization Considerations

The EMIFB memory controller has two reset signals, CHIP_RST and MOD_G_RST. The CHIP_RST is a module-level reset that resets both the state machine as well as the EMIFB memory controller memory-mapped registers. The MOD_G_RST resets the state machine only. If the EMIFB memory controller is reset independently of other peripherals, the user's software should not perform memory, as well as register accesses, while CHIP_RST or MOD_G_RST are asserted. If memory or register accesses are performed while the EMIFB memory controller is in the reset state, other masters may hang. Following the rising edge of CHIP_RST or MOD_G_RST, the EMIFB memory controller immediately begins its initialization sequence. Command and data stored in the EMIFB memory controller FIFOs are lost. Table 19 describes the different methods for asserting each reset signal. The Power and Sleep Controller (PSC) acts as a master controller for power management for all of the peripherals on the device. Figure 9 shows the EMIFB memory controller reset diagram.

Table 19. Reset Sources

Reset Signal	Reset Source
CHIP_RST	Hardware/device reset
MOD_G_RST	Power and sleep controller

Figure 9. EMIFB Memory Controller Reset Block Diagram



When the RESET pin on the device is asserted or a system reset is issued from Code Composer Studio, EMIFB memory controller's behavior is same as CHIP_RST assertion. In all these cases, the EMIFB will exit the reset state when the reset is released and after the PLL controller releases the entire device from reset. In all cases, EMIFB automatically begins running the SDRAM initialization sequence after coming out of reset. Even though the initialization procedure is automatic, a special procedure, found in Section 2.6.5 must still be followed.

2.8 Interrupt Support

EMIFB supports Line Trap Interrupt, which is caused by use of unsupported addressing mode. EMIFB supports only linear incrementing and cache line wrap addressing modes . If an access request for an unsupported addressing mode is received, the EMIFB will set the LT bit in the interrupt raw register (IRR) and treat the request as a linear incrementing request. For details on interrupt support, interrupt events, and interrupt multiplexing, see your device-specific *System Reference Guide*.

2.9 EDMA Event Support

EMIFB memory controller is a DMA slave peripheral and therefore does not generate DMA events. Data read and write requests may be made directly, by masters and the DMA.

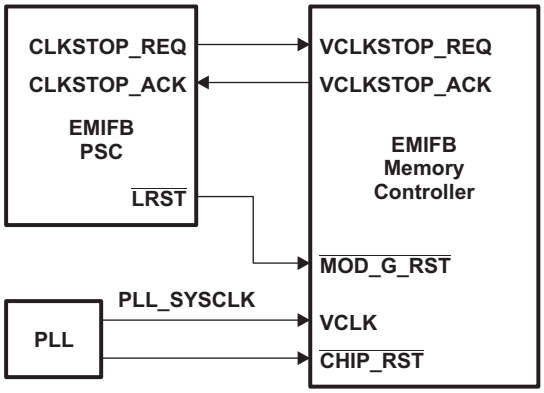
2.10 Power Management

Power dissipation from the EMIFB memory controller may be managed by two methods:

- Self-refresh mode (see Section 2.6.7)
- Power-down mode
- Gating input clocks to the module off.
- Power management in mobile SDRAM, using partial array self refresh.

Gating input clocks off to the EMIFB memory controller achieves higher power savings when compared to the power savings of self-refresh or power-down mode. The input clocks are turned off outside of the EMIFB memory controller through the use of the Power and Sleep Controller (PSC) and the PLL controller. Figure 10 shows the connections between the EMIFB memory controller, PSC, and PLL. Before gating clocks off, the EMIFB memory controller must place the SDR SDRAM memory in self-refresh mode by clearing the SR_PD bit to 0 and setting the LP_MODE bit to 1 in the SDRAM refresh control register (SDRFC). If the external memory requires a continuous clock, the EMIFB memory controller clock provided by PLL must not be turned off because this may result in data corruption. See the following subsections for the proper procedures to follow when stopping the EMIFB memory controller clocks.

Figure 10. EMIFB Memory Controller Power and Sleep Controller Diagram



2.10.1 Power Management Using Self-Refresh Mode

The EMIFB can be placed into a self-refresh state in order to place the attached SDRAM devices into self-refresh mode, which consumes less power for most SDRAM devices. In this state, the attached SDRAM device uses an internal clock to perform its own auto refresh cycles. This maintains the validity of the data in the SDRAM without the need for any external commands. Refer to Section 2.6.7 for more details on placing the EMIFB into the self-refresh state.

2.10.2 Power Management Using Power-Down Mode

In case of power-down, to lower the power consumption, EMIFB drives EMB_SDCKE low. EMB_SDCKE goes high when there is a need to send refresh (REFR) commands, after which EMB_SDCKE is again driven low. EMB_SDCKE remains low until any request arrives. Refer to Section 2.6.8 for more details on placing EMIFB in power-down mode.

2.10.3 Power Management Using Clock Stop

LPSC of EMIFB memory controller can be programmed to be in one of the following states:

- Enable
- Disable
- Auto sleep
- Auto wake
- Sync reset

Each of the states is described in the following sections.

2.10.3.1 LPSC Disable and Enable

To achieve maximum power savings VCLK, MCLK and EMB_CLK should be gated off. Perform the following procedure when shutting down clocks to achieve maximum power savings:

- EMIFB should be put to self-refresh mode before stopping the clock. Refer to Section 2.6.7 for details on self-refresh mode. The EMIFB memory controller will complete any outstanding accesses and backlogged refresh cycles and then place the EMIFB memory controller in self-refresh mode.
- To enable clock stopping, MCLKSTOP_EN bit in SDRFC must be set to 1. Refer to Section 3.3 for details.
- Then, program the LPSC of EMIFB to disable VCLK. For details on how to program the PSC, see your device-specific *System Reference Guide*.

Clocks should not be stopped while data transfer is in progress. Only after transfer is completed, clock stop request should be issued.

To turn clocks back on and start using EMIFB:

- Program the LPSC of EMIFB to enable VCLK.
- Clear MCLKSTOP_EN bit in SDRFC to 0.
- Bring EMIFB out of self-refresh mode. Refer to Section 2.6.7 for details on self-refresh mode.

2.10.3.2 LPSC Auto Sleep and Auto Wake

Apart from disable and enable, EMIFB memory controller can make use of auto sleep and auto wake facility. Following describes the procedure to be followed to put EMIFB memory controller in auto sleep state:

- EMIFB should be put to self-refresh mode before stopping the clock. Refer to Section 2.6.7 for details on self-refresh mode. The EMIFB memory controller will complete any outstanding accesses and backlogged refresh cycles and then place the 0EMIFB memory controller in self-refresh mode.
- To enable clock stopping, MCLKSTOP_EN bit in SDRFC must be set to 1. Refer to Section 3.3 for details.
- Then, program the LPSC of EMIFB for auto sleep, to gate off the clocks.

Register and memory access requests are honored while EMIFB is in auto sleep state. When EMIFB sees a request while it is in auto sleep state, it automatically returns to enable state, processes the request, and returns back to auto sleep state until further requests come.

On frequent requests, EMIFB switches between auto sleep and enable states. To bring EMIFB back to the enable state permanently, auto wake can be used. Following procedure is followed for performing auto wake.

- Program the LPSC of EMIFB for auto wake.
- Clear MCLKSTOP_EN bit in SDRFC to 0.
- Bring EMIFB out of self-refresh mode. Refer to Section 2.6.7 for details on self-refresh mode.

After auto wake, EMIFB is in enable state and clocks run continuously.

2.10.3.3 LPSC Sync Reset

Sync reset of EMIFB through LPSC doesn't reset the EMIFB registers or memory. Thus EMIFB LPSC sync reset acts similar to EMIFB LPSC disable. Following is the procedure to put EMIFB in sync reset state

- EMIFB should be put to self-refresh mode before stopping the clock. Refer to Section 2.6.7 for details on self-refresh mode. The EMIFB memory controller will complete any outstanding accesses and backlogged refresh cycles and then place the EMIFB memory controller in self-refresh mode.
- To enable clock stopping, MCLKSTOP_EN bit in SDRFC must be set to 1. Refer to Section 3.3 for details.
- Then, program the LPSC of EMIFB to reset state.

On sync reset, requests to EMIFB are not honored. To bring EMIFB back to enable state, use the enable procedure described in Section 2.10.3.1.

2.11 Emulation Considerations

EMIFB memory controller will remain fully functional during emulation halts, to allow emulation access to external memory.

3 Registers

The external memory interface (EMIFB) is controlled by programming its internal memory-mapped registers (MMRs). Table 20 lists the memory-mapped registers of the EMIFB memory controller.

Note: All EMIFB MMRs support only word, that is, 32-bit, accesses. Performing a byte (8-bit) or halfword (16-bit) write to these registers results in undefined behavior.

The EMIFB base controller registers must always be accessed using 32-bit accesses (unless otherwise specified in this document). For the base address of the memory-mapped registers of EMIFB, see your device-specific data manual.

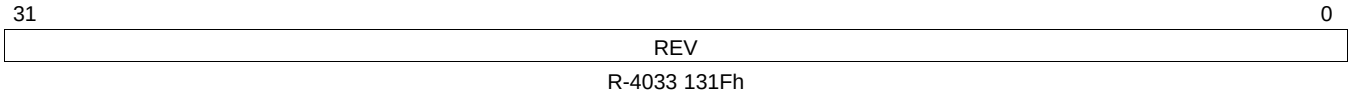
Table 20. EMIFB Base Controller Registers

Offset	Acronym	Register	Section
0h	REVID	Revision ID Register	Section 3.1
8h	SDCFG	SDRAM Configuration Register	Section 3.2
Ch	SDRFC	SDRAM Refresh Control Register	Section 3.3
10h	SDTIM1	SDRAM Timing 1 Register	Section 3.4
14h	SDTIM2	SDRAM Timing 2 Register	Section 3.5
1Ch	SDCFG2	SDRAM Configuration 2 Register	Section 3.6
20h	BPRIO	Peripheral Bus Burst Priority Register	Section 3.7
40h	PC1	Performance Counter 1 Register	Section 3.8
44h	PC2	Performance Counter 2 Register	Section 3.9
48h	PCC	Performance Counter Configuration Register	Section 3.10
4Ch	PCMRS	Performance Counter Master Region Select Register	Section 3.11
50h	PCT	Performance Counter Time Register	Section 3.12
C0h	IRR	Interrupt Raw Register	Section 3.13
C4h	IMR	Interrupt Mask Register	Section 3.14
C8h	IMSR	Interrupt Mask Set Register	Section 3.15
CCh	IMCR	Interrupt Mask Clear Register	Section 3.16

3.1 Revision ID Register (REVID)

This is read-only ID register of EMIFB. The REVID is shown in Figure 11 and described in Table 21.

Figure 11. Revision ID Register (REVID)



LEGEND: R = Read only; -*n* = value after reset

Table 21. Revision ID Register (REVID) Field Descriptions

Bit	Field	Value	Description
31-0	REV	4033 131Fh	Revision ID value of EMIFB.

3.2 SDRAM Configuration Register (SDCFG)

The SDRAM configuration register (SDCFG) is used to configure various parameters of the SDRAM controller such as the number of internal banks, the internal page size, and the CAS latency to match those of the attached SDRAM device. The SDCFG is shown in Figure 12 and described in Table 22.

BOOT_UNLOCK bit usage - The following sequence must be followed to change the value of the SDREN and MSDRAM_ENABLE bits.

1. Set the BOOT_UNLOCK bit to 1.
2. Write a 0 to the BOOT_UNLOCK bit along with the desired values for the SDREN/MSDRAM_ENABLE bits. The value of the bits is then updated.

TIMUNLOCK bit usage - The following sequence must be followed to change the value of any field affected by the TIMUNLOCK bit.

1. Write a 1 to the TIMUNLOCK bit along with the desired value for the CL field. The value of the CL field is then updated.
2. Update any of the fields required in the SDRAM timing registers (SDTIM1 and SDTIM2).
3. Clear the TIMUNLOCK bit to 0 to prevent any further changes.

Note: Writing to the lower two bytes of this register will cause the EMIF to start the SDRAM initialization sequence.

Figure 12. SDRAM Configuration Register (SDCFG)

31	27	26	25	24
Reserved		IBANK_POS	MSDRAM_ENABLE	Reserved
R-0		R/W-0	R/W-0	R-0

23	22	17	16
BOOT_UNLOCK	Reserved		SDREN
R/W-0	R-0		R/W-1

15	14	13	12	11	9	8
TIMUNLOCK	NM	Reserved		CL		Reserved
R/W-0	R/W-0	R-0		R/W-3h		R-0

7	6	4	3	2	0
Reserved	IBANK		EBANK	PAGESIZE	
R-0	R/W-2h		R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 22. SDRAM Configuration Register (SDCFG) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	All writes to these bit(s) must always have a value of 0.
26	IBANK_POS		Internal bank position. This bit is writeable only when the BOOT_UNLOCK bit is unlocked.
		0	Set to 0 to assign internal bank address bits from logical address as shown in Table 15 and Table 16. Set this bit to 0 when interfacing with SDR SDRAM.
		1	Set to 1 to assign internal bank address bits from logical address as shown in Table 17. Set this bit to 1 when interfacing with mobile SDRAM.
25	MSDRAM_ENABLE		Mobile SDRAM Enable. This bit is writeable only when the BOOT_UNLOCK bit is unlocked. For mobile SDR SDRAM, this bit is only valid when SDREN is set to 1.
		0	mSDR (mobile SDR) is disabled.
		1	When this bit is 1 and SDREN = 1, then mSDR is enabled.

Table 22. SDRAM Configuration Register (SDCFG) Field Descriptions (continued)

Bit	Field	Value	Description
24	Reserved	0	All writes to these bit(s) must always have a value of 0.
23	BOOT_UNLOCK		Boot unlock. Set to 1 to change the values of the fields that are affected by the BOOT_UNLOCK bit. See the description of usage of the BOOT_UNLOCK bit.
		0	The SDREN bit in this register may not be changed.
		1	The SDREN bit in this register may be changed.
22-17	Reserved	0	All writes to these bit(s) must always have a value of 0.
16	SDREN		SDRAM Enable. Active high bit which enables the SDRAM mode of the EMIFB controller. This bit is writeable only when the BOOT_UNLOCK bit is unlocked.
		0	SDRAM initialization and refreshes disabled, but SDRAM write/read transactions allowed. This bit must not be cleared to 0 when EMIFB is in self-refresh state.
		1	SDRAM fully enabled.
15	TIMUNLOCK		Timing unlock. Controls the write permission settings for the SDRAM timing register 1 (SDTIM1) and SDRAM timing register 2 (SDTIM2).
		0	CL bit in this register and register fields in SDTIM1 and SDTIM2 may not be changed.
		1	CL bit in this register and register fields in SDTIM1 and SDTIM2 may be changed.
14	NM		NM (Narrow mode). SDRAM data bus width. A write to this field will cause the EMIFB to start the SDRAM initialization sequence.
		0	32-bit SDR SDRAM
		1	16-bit SDR SDRAM
13-12	Reserved	0	All writes to these bit(s) must always have a value of 0.
11-9	CL	0-7h	CAS Latency. The value of this field defines the CAS latency to be used when accessing connected SDRAM devices. A write to this field will cause the EMIFB to start the SDRAM initialization sequence. This field is writeable only when the TIMUNLOCK bit is unlocked.
		0-1h	Reserved
		2h	CAS latency of 2
		3h	CAS latency of 3
		4h-7h	Reserved
8-7	Reserved	0	All writes to these bit(s) must always have a value of 0.
6-4	IBANK	0-7h	Internal SDRAM Bank setup. Defines number of banks inside connected SDRAM devices. A write to this field will cause the EMIFB to start the SDRAM initialization sequence.
		0	1 bank SDRAM devices
		1h	2 bank SDRAM devices
		2h	4 bank SDRAM devices
		3h-7h	Reserved
3	EBANK		External chip select setup. Always write 0 to this field. A write to this field will cause the EMIFB to start the SDRAM initialization sequence.
		0	Use EMB_CS for all SDRAM accesses.
		1	Reserved
2-0	PAGESIZE	0-7h	Page Size. Defines the internal page size of connected SDRAM devices. A write to this field will cause the EMIFB to start the SDRAM initialization sequence.
		0	256-word pages requiring 8 column address bits.
		1h	512-word pages requiring 9 column address bits.
		2h	1024-word pages requiring 10 column address bits.
		3h	2048-word pages requiring 11 column address bits.
		4h-7h	Reserved

3.3 SDRAM Refresh Control Register (SDRFC)

The SDRAM refresh control register (SDRFC) is used to configure the rate at which connected SDRAM devices will be automatically refreshed by the EMIFB. In addition, this register is used to put the attached SDRAM device into Self-Refresh/ Power-Down mode. The SDRFC is shown in Figure 13 and described in Table 23.

Figure 13. SDRAM Refresh Control Register (SDRFC)

31	30	29 24	23	22 16
LP_MODE	MCLKSTOP_EN	Reserved	SR_PD	Reserved
R/W-0	R/W-0	R-0	R/W-0	R-0

15 0
REFRESH_RATE
R/W-04E2h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 23. SDRAM Refresh Control Register (SDRFC) Field Descriptions

Bit	Field	Value	Description
31	LP_MODE		Low Power mode (Self Refresh). Writing a 1 to this bit will cause connected SDRAM devices to be place into self-refresh mode and the EMIFB to enter the self-refresh state.
		0	SDRAM is not to be placed in self-refresh/power-down mode.
		1	SDRAM placed in self-refresh/power-down mode depending on the value of SR_PD bit.
30	MCLKSTOP_EN		mclk Stop Enable. Writing a 1 to this bit enables mclk stopping.
		0	mclk stopping disabled.
		1	mclk stopping enabled.
29-24	Reserved	0	Reserved.
23	SR_PD		Self-refresh or power-down select. This bit is ignored when LP_MODE bit is cleared to 0.
		0	When LP_MODE = 1, clear this bit to 0 to cause connected SDRAM devices to be placed into self-refresh mode.
		1	When LP_MODE = 1, set this bit to 1 to cause connected SDRAM devices to be placed into power-down mode.
22-16	Reserved	0	Reserved.
15-0	REFRESH_RATE	0-FFFFh	Refresh Rate. Defines the rate at which connected SDRAM devices will be refreshed.
			SDRAM refresh rate = EMIF rate/REFRESH_RATE
			where EMIF rate is equal to EMIFB SDRAM clock rate.
			Writing a value < 0100h to this field causes it to be loaded with 2 × T_RFC value from SDRAM timing 1 register (SDTIM1).
			The required refresh rate is derived from the SDRAM device data sheet.

3.4 SDRAM Timing 1 Register (SDTIM1)

The SDRAM timing 1 register (SDTIM1) configures the SDRAM memory controller to meet many of the AC timing specification of the SDRAM memory. The SDTIM1 is programmable only when the TIMUNLOCK bit is set to 1 in the SDRAM configuration register (SDCFG). Note that EMB_CLK is equal to the period of the EMB_CLK signal. See the SDRAM memory data sheet for information on the appropriate values to program each field. The SDTIM1 is shown in Figure 14 and described in Table 24.

Figure 14. SDRAM Timing 1 Register (SDTIM1)

31–25	24–22	21–19	18–16
T_RFC	T_RP	T_RCD	T_WR
R/W-Ah	R/W-3h	R/W-3h	R/W-1h

15–11	10–6	5–3	2–0
T_RAS	T_RC	T_RRD	Reserved
R/W-7h	R/W-Ah	R/W-2h	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 24. SDRAM Timing 1 Register (SDTIM1) Field Descriptions

Bit	Field	Value	Description
31-25	T_RFC	0-7Fh	Specifies the minimum number of EMB_CLK cycles from a refresh or load mode command to a refresh or activate command, minus 1. Corresponds to the t_{rfc} AC timing parameter in the SDRAM data sheet. Calculate by: T_RFC = (t_{rfc} /EMB_CLK) - 1
24-22	T_RP	0-7h	Specifies the minimum number of EMB_CLK cycles from a precharge command to a refresh or activate command, minus 1. Corresponds to the t_{rp} AC timing parameter in the SDRAM data sheet. Calculate by: T_RP = (t_{rp} /EMB_CLK) - 1
21-19	T_RCD	0-7h	Specifies the minimum number of EMB_CLK cycles from an activate command to a read or write command, minus 1. Corresponds to the t_{rcd} AC timing parameter in the SDRAM data sheet. Calculate by: T_RCD = (t_{rcd} /EMB_CLK) - 1
18-16	T_WR	0-7h	Specifies the minimum number of EMB_CLK cycles from the last write transfer to a precharge command, minus 1. Corresponds to the t_{wr} AC timing parameter in the SDRAM data sheet. Calculate by: T_WR = (t_{wr} /EMB_CLK) - 1 When the value of this field is changed from its previous value, the initialization sequence will begin.
15-11	T_RAS	0-1Fh	Specifies the minimum number of EMB_CLK cycles from an activate command to a precharge command, minus 1. Corresponds to the t_{ras} AC timing parameter in the SDRAM data sheet. Calculate by: T_RAS = (t_{ras} /EMB_CLK) - 1 T_RAS must be greater than or equal to T_RCD.
10-6	T_RC	0-1Fh	Specifies the minimum number of EMB_CLK cycles from an activate command to an activate command, minus 1. Corresponds to the t_{rc} AC timing parameter in the SDRAM data sheet. Calculate by: T_RC = (t_{rc} /EMB_CLK) - 1
5-3	T_RRD	0-7h	Specifies the minimum number of EMB_CLK cycles from an activate command to an activate command in a different bank, minus 1. Corresponds to the t_{rrd} AC timing parameter in the SDRAM data sheet. Calculate by: T_RRD = (t_{rrd} /EMB_CLK) - 1 Note: for an 8 bank SDRAM device this field must be equal to $((4 \times t_{RRD}) + (2 \times t_{CK})) / (4 \times t_{CK}) - 1$.
2-0	Reserved	0	All writes to these bit(s) must always have a value of 0.

3.5 SDRAM Timing 2 Register (SDTIM2)

Like SDRAM timing 1 register (SDTIM1), the SDRAM timing register 2 (SDTIM2) also configures the SDRAM memory controller to meet the AC timing specification of the SDRAM memory. The SDTIM2 is programmable only when the TIMUNLOCK bit is set to 1 in the SDRAM configuration register (SDCFG). Note that EMB_CLK is equal to the period of the EMB_CLK signal. See the SDRAM data sheet for information on the appropriate values to program each field. SDTIM2 is shown in Figure 15 and described in Table 25.

Figure 15. SDRAM Timing 2 Register (SDTIM2)

31	30 27	26 23	22 16
Rsvd	T_RAS_MAX	Reserved	T_XSR
R-0	R/W-Eh	R-0	R/W-Ah

15 5	4 0
Reserved	T_CKE
R/W-0	R/W-7h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 25. SDRAM Timing 2 Register (SDTIM2) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	All writes to these bit(s) must always have a value of 0.
30-27	T_RAS_MAX	0-Fh	Maximum number of refresh_rate intervals from Activate to Precharge command.
26-23	Reserved	0	All writes to these bit(s) must always have a value of 0.
22-16	T_XSR	0-7Fh	Minimum number of EMB_CLK cycles from Self-Refresh exit to any command other than a Read command, minus one. This field must satisfy t_{XSR} for the SDRAM device. T_XSR = (t_{xsr} /EMIF_CLK) - 1
15-5	Reserved	0	All writes to these bit(s) must always have a value of 0.
4-0	T_CKE	0-1Fh	Minimum number of EMB_CLK cycles between EMB_SDCKE changes, minus one. This field must satisfy t_{RAS} for the SDRAM device. T_CKE = (t_{ras} /EMIF_CLK) - 1

3.6 SDRAM Configuration 2 Register (SDCFG2)

The SDRAM configuration 2 register (SDCFG2) helps programming the partial array self refresh feature of mobile SDRAM. SDCFG2 is shown in Figure 16 and described inTable 26.

Figure 16. SDRAM Configuration 2 Register (SDCFG2)

31	19	18	16
Reserved		PASR	
R-0		R/W-0	

15	3	2	0
Reserved		ROWSIZE	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 26. SDRAM Configuration 2 Register (SDCFG2) Field Description

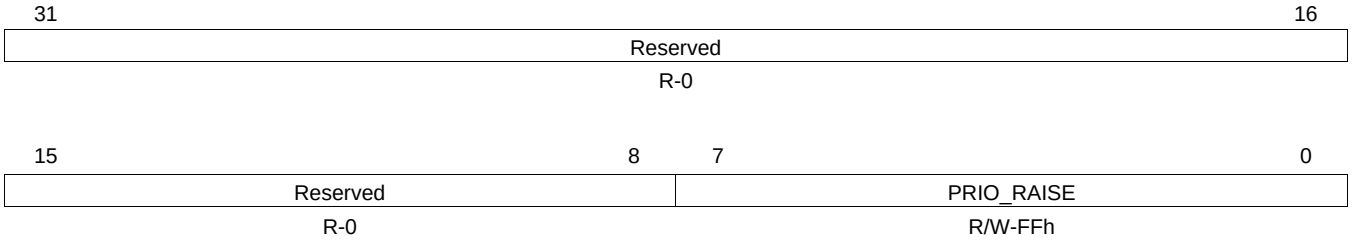
Bit	Field	Value	Description
31-19	Reserved	0	All writes to these bit(s) must always have a value of 0.
18-16	PASR	0-7h	Partial Array Self Refresh. These bits get loaded into the Extended Mode Register of a mobile SDRAM during initialization. A write to this field will cause the EMIFB to start the SDRAM initialization sequence.
		0	4 banks will be refreshed.
		1h	2 banks will be refreshed.
		2h	1 bank will be refreshed.
		3h-4h	Reserved.
		5h	1/2 bank will be refreshed.
		6h	1/4 bank will be refreshed.
		7h	Reserved.
15-3	Reserved	0	All writes to these bit(s) must always have a value of 0.
2-0	ROWSIZE	0-7h	Row Size. Defines the number of row address bits of connected mobile SDRAM devices. This field is only used when IBANK_POS bit in the SDRAM configuration register (SDCFG) is set to 1. A write to this field will cause the EMIFB to start the SDRAM initialization sequence. This bit applicable only when EMIFB controller is configured to interface to mobile SDRAM.
		0h	9 row address bits used.
		1h	10 row address bits used.
		2h	11 row address bits used.
		3h	12 row address bits used.
		4h	13 row address bits used.
		5h	14 row address bits used.
		6h-7h	Reserved

3.7 Peripheral Bus Burst Priority Register (BPRIO)

The peripheral bus burst priority register (BPRIO) helps prevent command starvation within the SDRAM memory controller. To avoid command starvation, the SDRAM memory controller momentarily raises the priority of the oldest command in the command FIFO after a set number of 32-bit transfers have been made on the external memory bus. The PRIO_RAISE bit sets the number of transfers that must be made before the SDRAM memory controller raises the priority of the oldest command. The BPRIO is shown in Figure 17 and described in Table 27.

Proper configuration of the BPRIO is critical to correct system operation. The EMIFB controller always prioritizes accesses to open rows as highest if there is any bank conflict regardless of master priority. This is done to allow most efficient utilization of the SDRAM. However, it could lead to excessive blocking of high priority masters. If the PRIO_RAISE bits are cleared to 00h, then the EMIFB controller always honors the master priority, regardless of open row/bank status. For most systems, the BPRIO should be set to a moderately low value to provide an acceptable balance of SDRAM efficiency and latency for high priority masters (for example, 10h or 20h).

Figure 17. Peripheral Bus Burst Priority Register (BPRIO)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 27. Peripheral Bus Burst Priority Register (BPRIO) Field Descriptions

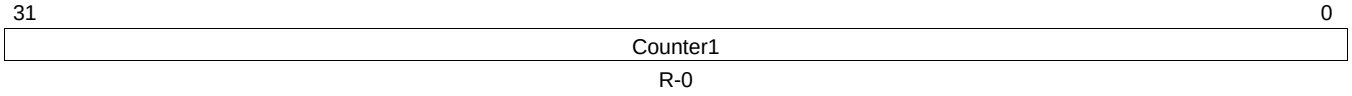
Bit	Field	Value	Description
31-8	Reserved	0	All writes to these bit(s) must always have a value of 0.
7-0	PRIO_RAISE	0-FFh	Priority raise old counter. Specifies the number of 32-bit memory transfers after which the SDRAM memory controller will elevate the priority of the oldest command in the command FIFO. Clearing to 00h will ensure master priority is strictly honored (at the cost of decreased EMIFB efficiency, as open row will always be closed immediately if any bank conflict occurs). Recommended setting for typical system operation is between 10h and 20h.

3.8 Performance Counter 1 Register (PC1)

For debug or gathering performance statistics, the PC1 and PC2 counters and associated configuration registers are provided. These are intended for debug and analysis only. By configuring the performance counter configuration register (PCC) to define the type of statistics to gather and configuring the performance counter master region select register (PCMRS) to filter accesses only to specific chip select regions, performing system applications and then reading these counters, different statistics can be gathered. To reset the counters, you must reset (SYNC RESET) the EMIFB module through the PSC (for details on the PSC, see your device-specific *System Reference Guide*.

The performance counter 1 register (PC1) is shown in Figure 18 and described in Table 28.

Figure 18. Performance Counter 1 Register (PC1)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

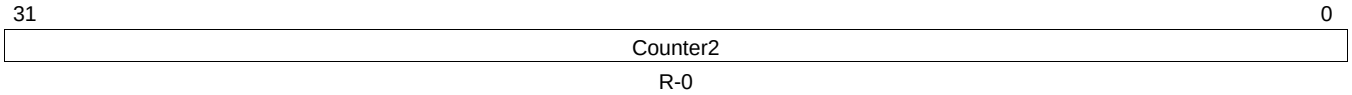
Table 28. Performance Counter 1 Register (PC1) Field Descriptions

Bit	Field	Value	Description
31-0	Counter1	0-FFFF FFFFh	32-bit counter that can be configured as specified in the performance counter configuration register (PCC) and the performance counter master region select register.

3.9 Performance Counter 2 Register (PC2)

The performance counter 2 register (PC2) is shown in Figure 19 and described in Table 29.

Figure 19. Performance Counter 2 Register (PC2)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 29. Performance Counter 2 Register (PC2) Field Descriptions

Bit	Field	Value	Description
31-0	Counter2	0-FFFF FFFFh	32-bit counter that can be configured as specified in the performance counter configuration register (PCC) and the performance counter master region select register.

3.10 Performance Counter Configuration Register (PCC)

The performance counter configuration register (PCC) is shown in Figure 20 and described in Table 30.

Table 31 shows the possible filter configurations for the two performance counters. These filter configurations can be used in conjunction with a Master ID and/or an external chip select to obtain performance statistics for a particular master and/or an external chip select.

Figure 20. Performance Counter Configuration Register (PCC)

31	30	29 20	19 16
CNTR2_MSTID_EN	CNTR2_REGION_EN	Reserved	CNTR2_CFG
R/W-0	R/W-0	R-0	R/W-1

15	14	13 4	3 0
CNTR1_MSTID_EN	CNTR1_REGION_EN	Reserved	CNTR1_CFG
R/W-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 30. Performance Counter Configuration Register (PCC) Field Descriptions

Bit	Field	Value	Description
31	CNTR2_MSTID_EN		Master ID filter enable for performance counter 2 register (PC2). Refer to Table 31 for details.
		0	Master ID filter is disabled. PC2 counts accesses from all masters to SDRAM.
		1	Master ID filter is enabled. PC2 counts accesses from the master, corresponding to the Master ID value in the MST_ID2 bit field of the performance counter master region select register (PCMRS).
30	CNTR2_REGION_EN		Chip select filter enable for performance counter 2 register (PC2). Refer to Table 31 for details.
		0	Chip select filter is disabled. PC2 counts total number of accesses (SDRAM + EMIFB memory-mapped register accesses). The REGION_SEL2 bit field value in the performance counter master region select register (PCMRS) is a don't care.
		1	Chip select filter is enabled. If the REGION_SEL2 bit field value in the performance counter master region select register (PCMRS) is:
			REGION_SEL2 = 0: PC2 counts accesses to SDRAM memory.
			REGION_SEL2 = 7h: PC2 counts accesses to EMIFB memory-mapped registers.
29-20	Reserved	0	Any writes to these bit(s) must always have a value of 0.
19-16	CNTR2_CFG	0-Fh	Filter configuration for performance counter 2 register (PC2). Refer to Table 31 for details.
15	CNTR1_MSTID_EN		Master ID filter enable for performance counter 1 register (PC1). Refer to Table 31 for details.
		0	Master ID filter is disabled. PC1 counts accesses from all masters to SDRAM.
		1	Master ID filter is enabled. PC1 counts accesses from the master, corresponding to the Master ID value in the MST_ID1 bit field of the performance counter master region select register (PCMRS).
14	CNTR1_REGION_EN		Chip select filter enable for performance counter 1 register (PC1). Refer to Table 31 for details.
		0	Chip select filter is disabled. PC1 counts total number of accesses (SDRAM + EMIFB memory-mapped register accesses). The REGION_SEL1 bit field value in the performance counter master region select register (PCMRS) is a don't care.
		1	Chip select filter is enabled. If the REGION_SEL1 bit field value in the performance counter master region select register (PCMRS) is:
			REGION_SEL1 = 0: PC1 counts accesses to SDRAM memory.
			REGION_SEL1 = 7h: PC1 counts accesses to EMIFB memory-mapped registers.
13-4	Reserved	0	Any writes to these bit(s) must always have a value of 0.
3-0	CNTR1_CFG	0-Fh	Filter configuration for performance counter 1 register (PC1). Refer to Table 31 for details.

Table 31. Performance Counter Filter Configuration

Performance Counter Configuration Register (PCC) Bit			
CNTR*n*_CFG	**CNTR*n*_REGION_EN**	**CNTR*n*_MSTID_EN**	**Description**
0	0	0 or 1	**Counts the total number of READ/WRITE commands the external memory controller receives.** The size of counter increments are determines by the size of the transfer and the default burst size (DBS). The counter breaks up transfers into sizes according to DBS. Therefore, counter increments for transfers aligned to DBS are equal to the transfer size divided by the DBS.
1h	0	0 or 1	**Counts the total number of ACTIVATE commands the external memory controller issues to SDRAM memory.** The counter increments by a value of 1 for every request to read/write data to a closed bank in SDRAM memory by the external memory controller.
2h	0 or 1	0 or 1	**Counts the total number of READ commands (read accesses) the EMIFB receives.** Counter increments for transfers aligned to the default burst size (DBS) are equal to the transfer size divided by the DBS.
3h	0 or 1	0 or 1	**Counts the total number of WRITE commands the EMIFB receives.** Counter increments for transfers aligned to the default burst size (DBS) are equal to the transfer size of data written to the DDR2 memory controller divided by the DBS.
4h	0	0	**Counts the number of external memory controller cycles (EMB_CLK cycles) that the command FIFO is full.** Use the following to calculate the counter value as a percentage: *% = counter value / total EMB_CLK cycles in a sample period* As the value of this counter approaches 100%, the EMIFB memory controller is approaching a congestion point where the command FIFO is full 100% of the time and a command will have to wait at the SCR to be accepted in the command FIFO.
5h-7h	0	0	Reserved
8h	0 or 1	0 or 1	**Counts the number of commands (requests) in the command FIFO that require a priority elevation.** To avoid command starvation, the EMIFB memory controller can momentarily raise the priority of the oldest command in the command FIFO after a set number of transfers have been made. The PRIO_RAISE bit field in the peripheral bus burst priority register (BPRIO) sets the number of the transfers that must be made before the EMIFB memory controller will raise the priority of the oldest command.
9h	0	0	**Counts the number of EMIFB memory controller cycles (EMB_CLK cycles) that a command is pending in the command FIFO.** This counter increments every cycle the command FIFO is not empty. Use the following to calculate the counter value as a percentage: *% = counter value / total EMB_CLK cycles in sample period* As the value of this counter approaches 100%, the number of cycles the EMIFB has a command in the command FIFO to service approaches 100%.
Ah-Fh	0	0	Reserved

3.11 Performance Counter Master Region Select Register (PCMRS)

The performance counter master region select register (PCMRS) is shown in Figure 21 and described in Table 32.

Figure 21. Performance Counter Master Region Select Register (PCMRS)

31	24	23	20	19	16
MST_ID2		Reserved		REGION_SEL2	
R/W-0		R-0		R/W-0	

15	8	7	4	3	0
MST_ID1		Reserved		REGION_SEL1	
R/W-0		R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

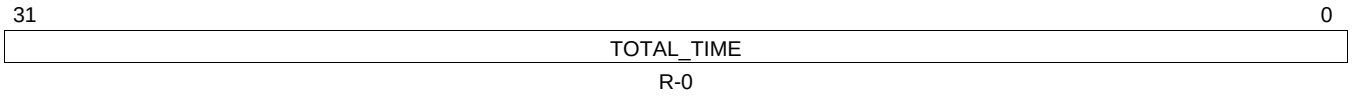
Table 32. Performance Counter Master Region Select Register (PCMRS) Field Descriptions

Bit	Field	Value	Description
31-24	MST_ID2	0-FFh	Master ID for performance counter 2 register (PC2). For the Master ID value for master peripherals in the device, see your device-specific *System Reference Guide*.
23-20	Reserved	0	Any writes to these bit(s) must always have a value of 0.
19-16	REGION_SEL2	0-Fh	Region select for performance counter 2 register (PC2).
		0	PC2 counts total SDRAM accesses.
		1h-6h	Reserved
		7h	PC2 counts total EMIFB memory-mapped register accesses.
		8h-Fh	Reserved
15-8	MST_ID1	0-FFh	Master ID for performance counter 1 register (PC1). For the Master ID value for master peripherals in the device, see your device-specific *System Reference Guide*.
7-4	Reserved	0	Any writes to these bit(s) must always have a value of 0.
3-0	REGION_SEL1	0-Fh	Region select for performance counter 1 register (PC1).
		0	PC1 counts total SDRAM accesses.
		1h-6h	Reserved
		7h	PC1 counts total EMIFB memory-mapped register accesses.
		8h-Fh	Reserved

3.12 Performance Counter Time Register (PCT)

The performance counter time register (PCT) is shown in Figure 22 and described in Table 33.

Figure 22. Performance Counter Time Register (PCT)



LEGEND: R = Read only; -*n* = value after reset

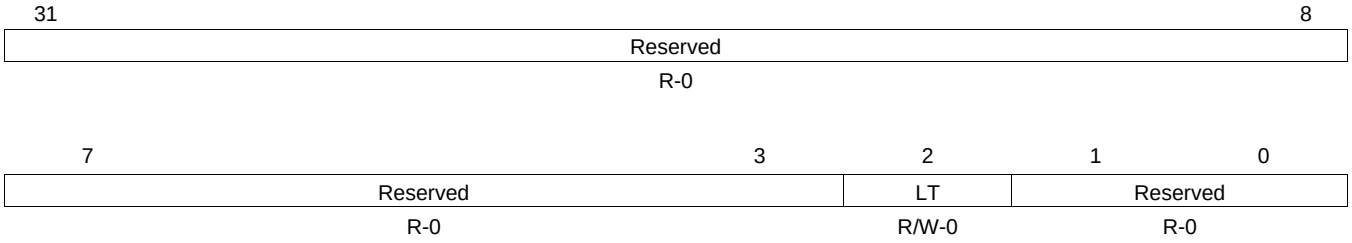
Table 33. Performance Counter Time Register (PCT) Field Description

Bit	Field	Value	Description
31-0	TOTAL_TIME	0-FFFF FFFFh	32-bit counter that continuously counts number for EMB_CLK cycles elapsed after EMIFB is brought out of reset.

3.13 Interrupt Raw Register (IRR)

The interrupt raw register (IRR) displays the raw status of the interrupt. If the interrupt condition occurs, the corresponding bit in IRR is set independent of whether or not the interrupt is enabled. The IRR is shown in Figure 23 and described in Table 34.

Figure 23. Interrupt Raw Register (IRR)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

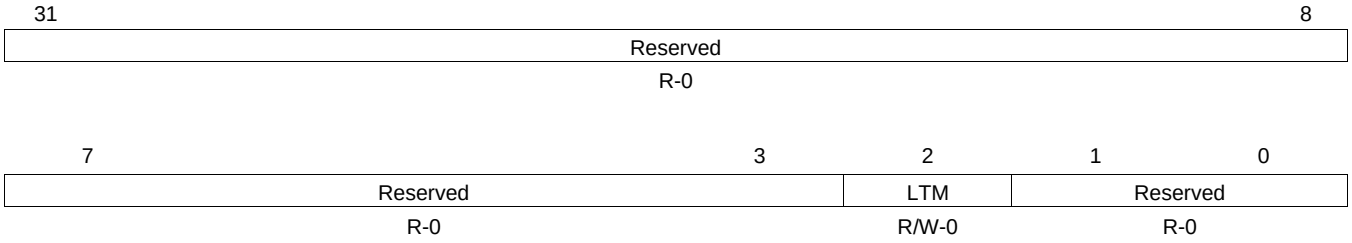
Table 34. Interrupt Raw Register (IRR) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	All writes to these bit(s) must always have a value of 0.
2	LT		Line Trap. Set to 1 by hardware to indicate illegal memory access type. Writing a 1 will clear this bit as well as the LTM bit in the interrupt mask register (IMR). Writing a 0 has no effect.
		0	Line trap hasn't occurred.
		1	Line trap has occurred due to use of unsupported addressing mode. EMIFB supports linear incrementing and cache line wrap addressing modes.
1-0	Reserved	0	All writes to these bit(s) must always have a value of 0.

3.14 Interrupt Mask Register (IMR)

The interrupt mask register (IMR) displays the status of the interrupt when it is enabled. If the interrupt condition occurs and the corresponding bit in the interrupt mask set register (IMSR) is set, then the IMR bit is set. The IMR bit is not set if the interrupt is not enabled in IMSR. The IMR is shown in Figure 24 and described in Table 35.

Figure 24. Interrupt Mask Register (IMR)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

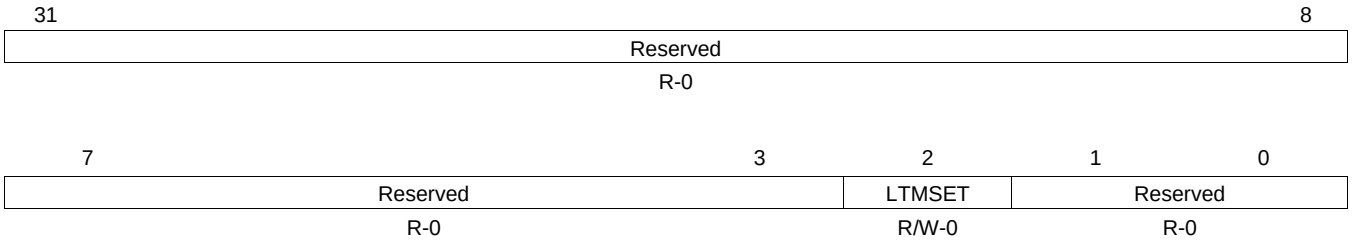
Table 35. Interrupt Mask Register (IMR) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	All writes to these bit(s) must always have a value of 0.
2	LTM		Masked Line Trap. Set to 1 by hardware to indicate illegal memory access type, only if the LTMSET bit in the interrupt mask set register (IMSR) is set to 1. Writing a 1 will clear this bit as well as the LT bit in the interrupt raw register (IRR). Writing a 0 has no effect.
		0	Line trap has not occurred.
		1	Line trap occurred due to use of unsupported addressing mode (only set if the LTMSET bit in IMSR is set).
1-0	Reserved	0	All writes to these bit(s) must always have a value of 0.

3.15 Interrupt Mask Set Register (IMSR)

The interrupt mask set register (IMSR) enables the memory controller interrupt. The IMSR is shown in Figure 25 and described in Table 36.

Figure 25. Interrupt Mask Set Register (IMSR)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

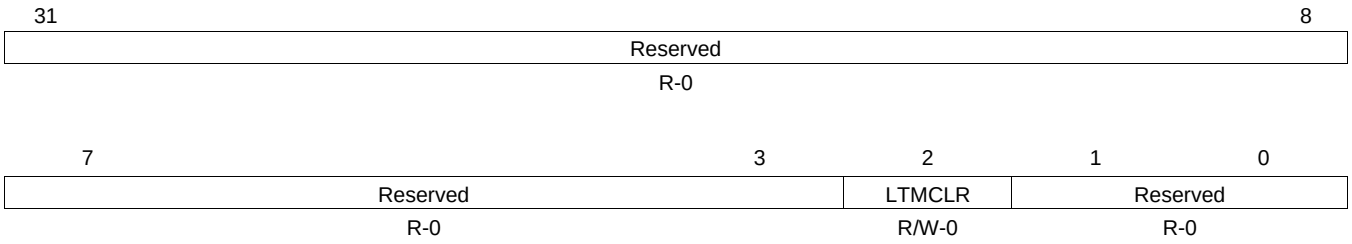
Table 36. Interrupt Mask Set Register (IMSR) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	All writes to these bit(s) must always have a value of 0.
2	LTMSET		Mask set for LTM bit in the interrupt mask register (IMR). Writing a 1 will enable the interrupt, and set this bit as well as the LTMCLR bit in the interrupt mask clear register (IMCR). The interrupt will not be enabled, and this bit as well as the LTMCLR bit will not be set if a 1 is written to this bit and the LTMCLR bit at the same time. Writing a 0 has no effect.
		0	Line trap interrupt is not enabled; a write of 1 to the LTMCLR bit in IMCR occurred.
		1	Line trap interrupt is enabled.
1-0	Reserved	0	All writes to these bit(s) must always have a value of 0.

3.16 Interrupt Mask Clear Register (IMCR)

The interrupt mask clear register (IMCR) disables the memory controller interrupt. Once an interrupt is enabled, it may be disabled by writing a 1 to the IMCR bit. The IMCR is shown in Figure 26 and described in Table 37.

Figure 26. Interrupt Mask Clear Register (IMCR)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 37. Interrupt Mask Clear Register (IMCR) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	All writes to these bit(s) must always have a value of 0.
2	LTMCLR		Mask clear for LTM bit in the interrupt mask register (IMR). Writing a 1 will disable the interrupt, and clear this bit as well as the LTMSET bit in the interrupt mask set register (IMSR). Writing a 0 has no effect.
		0	Line trap interrupt is not enabled.
		1	Line trap interrupt is enabled; a write of 1 to the LTMSET bit in IMSR occurred.
1-0	Reserved	0	All writes to these bit(s) must always have a value of 0.

Appendix A Example Configuration

The EMIFB memory controller allows a high degree of programmability for shaping SDRAM accesses. The programmability inherent to the EMIFB memory controller provides the EMIFB memory controller with the flexibility to interface with a variety of SDRAM devices. By programming the SDRAM configuration register (SDCFG), SDRAM refresh control register (SDRFC), SDRAM timing register 1 (SDTIM1), and SDRAM timing register 2 (SDTIM2), the EMIFB memory controller can be configured to meet the data sheet specification for JESD21-C compliant SDR SDRAM. This section presents an example describing how to interface the EMIFB memory controller to a JESD21-C SDR SDRAM 64MB device. The EMIFB memory controller is assumed to be operating at 133 MHZ.

A.1 Hardware Configuration

The following figures show how to connect the EMIFB memory controller to an SDR SDRAM device. Figure A-1 displays a 32-bit interface; therefore, two 16-bit SDR SDRAM devices are connected to the EMIFB memory controller. From Figure A-1, you can see that the data bus and data mask (byte enable) signals are point-to-point where as all other address, control, and clocks are not. Figure A-2 displays a 16-bit interface; therefore, all signals are point-to-point.

Figure A-1. Connecting EMIFB Memory Controller for 32-bit Connection

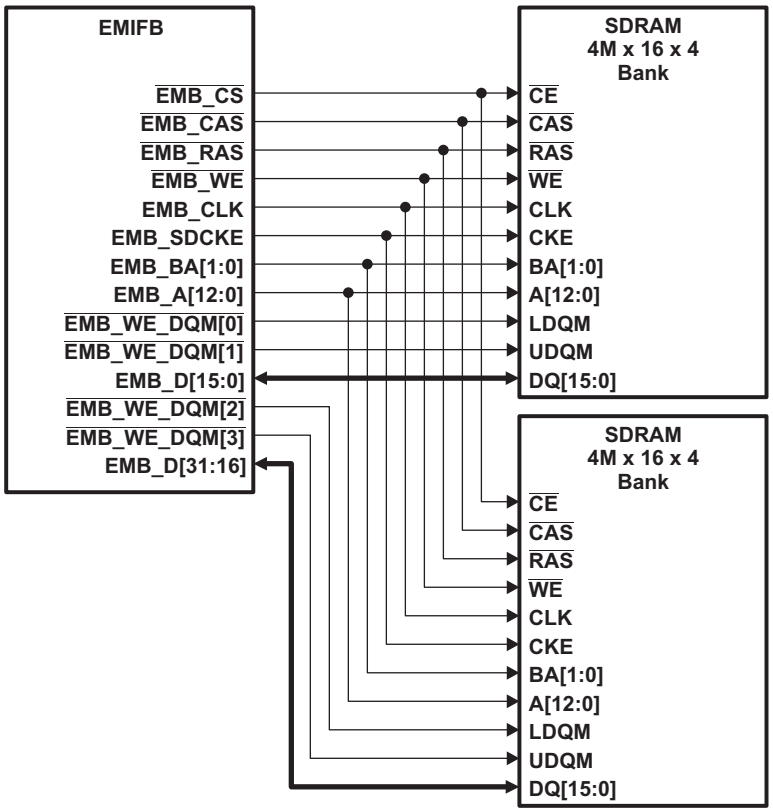
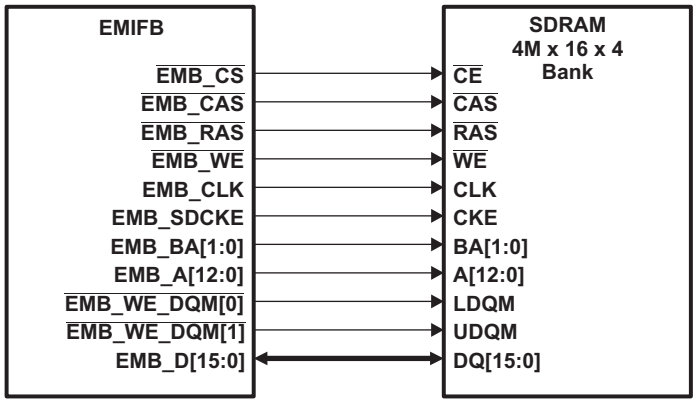


Figure A-2. Connecting EMIFB Memory Controller for 16-bit Connection



A.2 Software Configuration

Four memory-mapped registers must be programmed to configure the EMIFB memory controller to meet the data sheet specification of the attached SDR SDRAM device. The registers are:

- SDRAM configuration register (SDCFG)
- SDRAM refresh control register (SDRFC)
- SDRAM timing register 1 (SDTIM1)
- SDRAM timing register 2 (SDTIM2)

The following sections describe how to configure each of these registers. See Section 3 for more information on the EMIFB memory controller registers.

A.2.1 PLL Programming for EMIFB

The device PLL Controller should first be programmed to select the desired EMB_CLK frequency. Before doing this, the SDRAM should be placed into Self-Refresh Mode by setting the SR_PD bit and LP_MODE bit in SDRFC to 0 & 1 respectively. The EMB_CLK frequency can now be adjusted to the desired value by programming the appropriate SYSCLK domain of the PLL Controller. Once the PLL has been reprogrammed, remove the SDRAM from Self-Refresh by clearing the LP_MODE bit in SDRFC.

A.2.2 Configuring SDRAM Configuration Register (SDCFG)

The SDRAM configuration register (SDCFG) contains register fields that configure the EMIFB memory controller to match the data bus width, CAS latency, number of banks, and page size of the attached SDRAM memory. In this example, we assume the following configuration:

- Data bus width = 32 bits
- CAS latency = 2
- Number of banks = 4
- Page size = 512 words

Table A-1shows the resulting SDCFG configuration. Note that the value of the TIMUNLOCK field is dependent on whether or not it is desirable to unlock SDTIM1 and SDTIM2. The TIMUNLOCK bit should only be set to 1 when the SDTIM1 and SDTIM2 need to be updated.

Table A-1. SDCFG Configuration

Field	Value	Function Selection
TIMUNLOCK	x	Set to 1 to unlock the SDRAM timing register 1 (SDTIM1) and the SDRAM timing register 2 (SDTIM2). Cleared to 0 to lock SDTIM1 and SDTIM2.
NM	0	To configure the EMIFB memory controller for a 32-bit data bus width.
CL	2h	To select a CAS latency of 2.
IBANK	2h	To select 4 internal SDR SDRAM banks.
PAGESIZE	1h	To select 512-word page size.

A.2.3 Configuring SDRAM Refresh Control Register (SDRFC)

The SDRAM refresh control register (SDRFC) configures the EMIFB memory controller to meet the refresh requirements of the attached SDRAM device. SDRFC also allows the EMIFB memory controller to enter and exit self-refresh and power-down and enable and disable the MCLK stopping. In this example, we assume that the EMIFB memory controller is not is in self-refresh/power-down mode and that MCLK stopping is disabled. The REFRESH_RATE field in SDRFC is defined as the rate at which the attached SDRAM device is refreshed in SDRAM cycles.

The value of this field may be calculated using the following equation:

REFRESH_RATE = SDRAM clock frequency $\times$ SDRAM refresh rate

Assuming 64 ms(tREF), 8192 rows (2^{13}; 13 address lines), SDRAM refresh rate = 64/8192 = 7.8 µs.

Therefore, the following results assuming 133-MHZ SDRAM clock frequency.

REFRESH_RATE = 133 MHZ $\times$ 7.8 µs = 1037.4 Therefore, REFRESH_RATE = 1038 = 40Eh.

Table A-2 shows the resulting SDRFC configuration.

Table A-2. SDRFC Configuration

Field	Value	Function Selection
LP_MODE	0	EMIFB memory controller not put in low power mode.
MCLKSTOP_EN	0	MCLK stopping is disabled.
SR_PD	0	This bit is ignored when LP_MODE=0.
REFRESH_RATE	40Eh	Set to 40Eh SDRAM clock cycles to meet the SDRAM memory refresh rate requirement.

A.2.4 Configuring SDRAM Timing Registers (SDTIM1 and SDTIM2)

The SDRAM timing register 1(SDTIM1) and SDRAM timing register 2 (SDTIM2) configure the EMIFB memory controller to meet the data sheet timing parameters of the attached SDRAM device. Each field in SDTIM1 and SDTIM2 corresponds to a timing parameter in the SDRAM data sheet specification. Table A-3 and Table A-4 display the register field name and corresponding SDRAM data sheet parameter name along with the data sheet value. These tables also provide a formula to calculate the register field value and displays the resulting calculation. Each of the equations include a minus 1 because the register fields are defined in terms of SDRAM clock cycles minus 1. See Section 3.4and Section 3.5 for more information.

Table A-3. SDTIM1 Configuration

Register Field Name	SDRAM Data Manual Parameter Name	Description	Data Manual Value (ns)	Formula (Register field must be ≥)	Register Value
T_RFC	t_{RFC}	refresh cycle time	66	(t_{RFC}× f_{EMB_CLK}) - 1	8
T_RP	t_{RP}	precharge command to refresh or activate command	20	(t_{RP}× f_{EMB_CLK}) - 1	2
T_RCD	t_{RCD}	activate command to read/write command	20	(t_{RCD}× f_{EMB_CLK}) - 1	2
T_WR	t_{WR}	write recovery time	15	(t_{WR}× f_{EMB_CLK}) - 1	1
T_RAS	t_{RAS}	active to precharge command	44	(t_{RAS}× f_{EMB_CLK}) - 1	5
T_RC	t_{RC}	activate to activate command in the same bank	66	(t_{RC}× f_{EMB_CLK}) - 1	8
T_RRD	t_{RRD}	activate to activate command in a different bank	15	(t_{RRD}× f_{EMB_CLK}) - 1	1

Table A-4. SDTIM2 Configuration

Register Field Name	SDRAM Data Manual Parameter Name	Description	Data Manual Value (ns)	Formula (Register field must be ≥)	Register Value
T_RAS_MAX	t_{RAS_MAX}	refresh cycle time	66	(t_{RAS_MAX}/ SDRAM refresh rate) - 1	8
T_XSR	t_{XSR}	self refresh exit to any command other than a read command	75	(t_{XSR}× f_{EMB_CLK}) - 1	9
T_CKE	t_{CKE}	number of clock cycles between EMB_CKE changes	38	(t_{CKE}× f_{EMB_CLK}) - 1	5

Appendix B Revision History

Table B-1 lists the changes made since the previous version of this document.

Table B-1. Document Revision History

Reference	Additions/Modifications/Deletions
Section 2.6.13.2	Deleted Note.
Table 24	Changed formulas.
Section 3.5	Added third sentence.
Table 25	Changed formulas.
Section 3.7	Changed second paragraph.
Table 27	Deleted third sentence in Description.
Section 3.8	Changed last sentence in first paragraph.
Figure 20	Changed name of bit 15.
	Changed name of bit 14.
Table 30	Changed table.
Table 31	Changed table.
Figure 21	Changed figure.
Table 32	Changed table.

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 TI 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. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

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