TMS320C674x/OMAP-L1x Processor External Memory Interface A (EMIFA)

User's Guide

Literature Number: SPRUFL6F
December 2010

List of Figures

List of Tables

Read This First

About This Manual

This document describes the operation of the external memory interface A (EMIFA).

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.

SPRUGM5 — ***TMS320C6742 DSP System Reference Guide.*** Describes the C6742 DSP subsystem, system memory, device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, and system configuration module.

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.

SPRUGM6 — ***TMS320C6746 DSP System Reference Guide.*** Describes the C6746 DSP subsystem, system memory, device clocking, phase-locked loop controller (PLLC), power and sleep controller (PSC), power management, and system configuration module.

SPRUGJ7 — ***TMS320C6748 DSP System Reference Guide.*** Describes the C6748 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.

SPRUGM7 — ***OMAP-L138 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 A (EMIFA)

1 Introduction

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

The EMIFA SDRAM interface is not supported on all devices, see your device-specific data manual to see if the EMIFA SDRAM is supported on your device.

1.1 Purpose of the Peripheral

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

- Single data rate (SDR) SDRAM
- Asynchronous devices including NOR Flash, NAND Flash, and SRAM

The most common use for the EMIFA is to interface with both a flash device and an SDRAM device simultaneously. Appendix A contains an example of operating the EMIFA in this configuration.

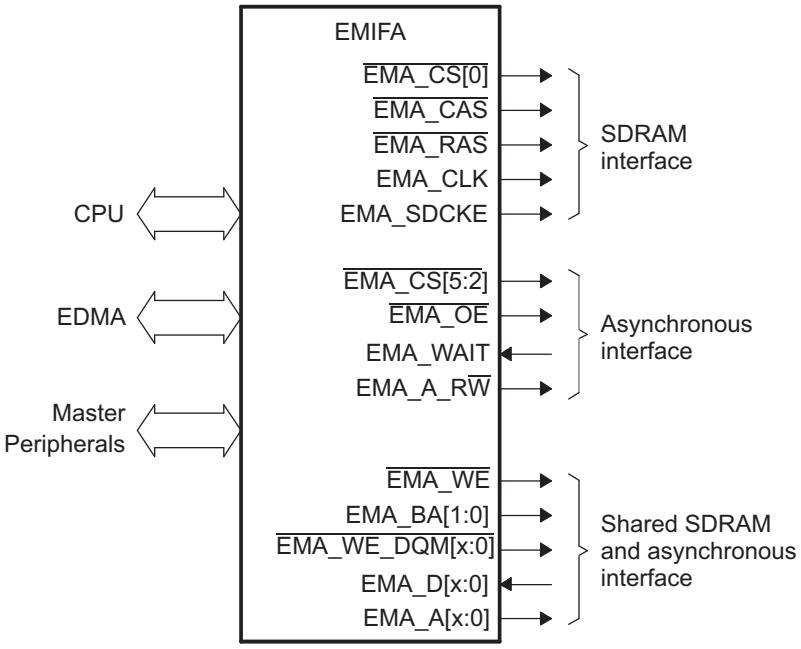
1.2 Features

The EMIFA includes many features to enhance the ease and flexibility of connecting to external SDR SDRAM and asynchronous devices. For details on features of EMIFA, see your device-specific data manual.

1.3 Functional Block Diagram

Figure 1 illustrates the connections between the EMIFA and its internal requesters, along with the external EMIFA pins. Section 2.2 contains a description of the entities internal to the SoC that can send requests to the EMIFA, along with their prioritization. Section 2.3 describes the EMIFA external pins and summarizes their purpose when interfacing with SDRAM and asynchronous devices.

Figure 1. EMIFA Functional Block Diagram



2 Architecture

This section provides details about the architecture and operation of the EMIFA. Both, SDRAM and asynchronous interface are covered, along with other system-related issues such as clock control and pin multiplexing.

The EMIFA SDRAM interface is not supported on all devices, see your device-specific data manual to see if the EMIFA SDRAM is supported on your device.

2.1 Clock Control

The EMIFA clock is output on the EMA_CLK pin and should be used when interfacing to external memories. The EMIFA clock (EMA_CLK) does not run during device reset. When the RESET pin is released and after the PLL controller releases the device from reset, EMA_CLK begins to oscillate at a frequency determined by the PLL controller.

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

2.2 EMIFA Requests

Different sources within the SoC can make requests to the EMIFA. These requests consist of accesses to SDRAM memory, asynchronous memory, and EMIFA registers. Because the EMIFA can process only one request at a time, a high performance crossbar switch exists within the SoC to provide prioritized requests from the different sources to the EMIFA. The sources are:

1. CPU
2. EDMA
3. Other master peripherals, like HPI, etc.

If a request is submitted from two or more sources simultaneously, the crossbar switch will forward the highest priority request to the EMIFA first. Upon completion of a request, the crossbar switch again evaluates the pending requests and forwards the highest priority pending request to the EMIFA.

When the EMIFA receives a request, it may or may not be immediately processed. In some cases, the EMIFA will perform one or more auto refresh cycles before processing the request. For details on the EMIFA's internal arbitration between performing requests and performing auto refresh cycles, see Section 2.12.

2.3 Pin Descriptions

This section describes the function of each of the EMIFA pins.

Table 1. EMIFA Pins Used to Access Both SDRAM and Asynchronous Memories

Pins(s)	I/O	Description
EMA_D[x:0]	I/O	**EMIFA data bus.** The number of available data bus pins varies among devices. See your device-specific data manual for details.
EMA_ A[x:0]	O	**EMIFA address bus.** When interfacing to an SDRAM device, these pins are primarily used to provide the row and column address to the SDRAM. The mapping from the internal program address to the external values placed on these pins can be found in Table 13. EMA_A[10] is also used during the PRE command to select which banks to deactivate. When interfacing to an asynchronous device, these pins are used in conjunction with the EMA_BA pins to form the address that is sent to the device. The mapping from the internal program address to the external values placed on these pins can be found in Section 2.5.1. The number of available address pins varies among devices. See your device-specific data manual for details.
EMA_BA[1:0]	O	**EMIFA 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 inTable 13. When interfacing to an asynchronous device, these pins are used in conjunction with the EMA_A pins to form the address that is sent to the device. The mapping from the internal program address to the external values placed on these pins can be found in Section 2.5.1.
$\overline{\text{EMA\_WE\_DQM[x:0]}}$	O	**Active-low 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. When interfacing to an asynchronous device, these pins are connected to byte enables. See Section 2.5 for details.
$\overline{\text{EMA\_WE}}$	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. When interfacing to an asynchronous device, this pin provides a signal which is active-low during the strobe period of an asynchronous write access cycle.

Table 2. EMIFA Pins Specific to SDRAM

Pin(s)	I/O	Description
$\overline{\text{EMA\_CS[0]}}$	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 EMIFA keeps this SDRAM chip select active, even if the EMIFA is not interfaced with an SDRAM device. This pin is deactivated when accessing the asynchronous memory bank and is reactivated on completion of the asynchronous assess.
$\overline{\text{EMA\_RAS}}$	O	**Active-low row address strobe pin.** This pin is connected to the $\overline{\text{RAS}}$ pin of the attached SDRAM device and is used for sending commands to the device.
$\overline{\text{EMA\_CAS}}$	O	**Active-low column address strobe pin.** This pin is connected to the $\overline{\text{CAS}}$ pin of the attached SDRAM device and is used for sending commands to the device.
EMA_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 2.4.7 for details.
EMA_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.

Table 3. EMIFA Pins Specific to Asynchronous Memory

Pin(s)	I/O	Description
EMA_CS[5:2]	O	**Active-low chip enable pins for asynchronous devices.** These pins are meant to be connected to the chip-select pins of the attached asynchronous device. These pins are active only during accesses to the asynchronous memory.
EMA_WAIT	I	**Wait input with programmable polarity / NAND Flash ready input.** A connected asynchronous device can extend the strobe period of an access cycle by asserting the EMA_WAIT input to the EMIFA as described in Section 2.5.7. To enable this functionality, the EW bit in the asynchronous 1 configuration register (CE2CFG) must be set to 1. In addition, the WP0 bit in CE2CFG must be configured to define the polarity of the EMA_WAIT pin. When the CS2NAND/CS3NAND/CS4NAND/CS5NAND bit in the NAND Flash control register (NANDFCR) is set, this pin instead functions as a NAND Flash ready input.
EMA_OE	O	**Active-low pin enable for asynchronous devices.** This pin provides a signal which is active-low during the strobe period of an asynchronous read access cycle.
EMA_A_RW	O	**EMIFA asynchronous read/write control.** This pin stays high during reads and stays low during writes (same duration as CS).

2.4 SDRAM Controller and Interface

The EMIFA can gluelessly interface to most standard SDR SDRAM devices and supports 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 EMIFA to perform read and write operations to externally connected SDR SDRAM devices. Also, Appendix A provides a detailed example of interfacing the EMIFA to a common SDRAM device.

2.4.1 SDRAM Commands

The EMIFA supports the SDRAM commands described in Table 4. Table 5 shows the truth table for the SDRAM commands, and an example timing waveform of the PRE command is shown in Figure 2. EMA_A[10] is pulled low in this example to deactivate only the bank specified by the EMA_BA pins.

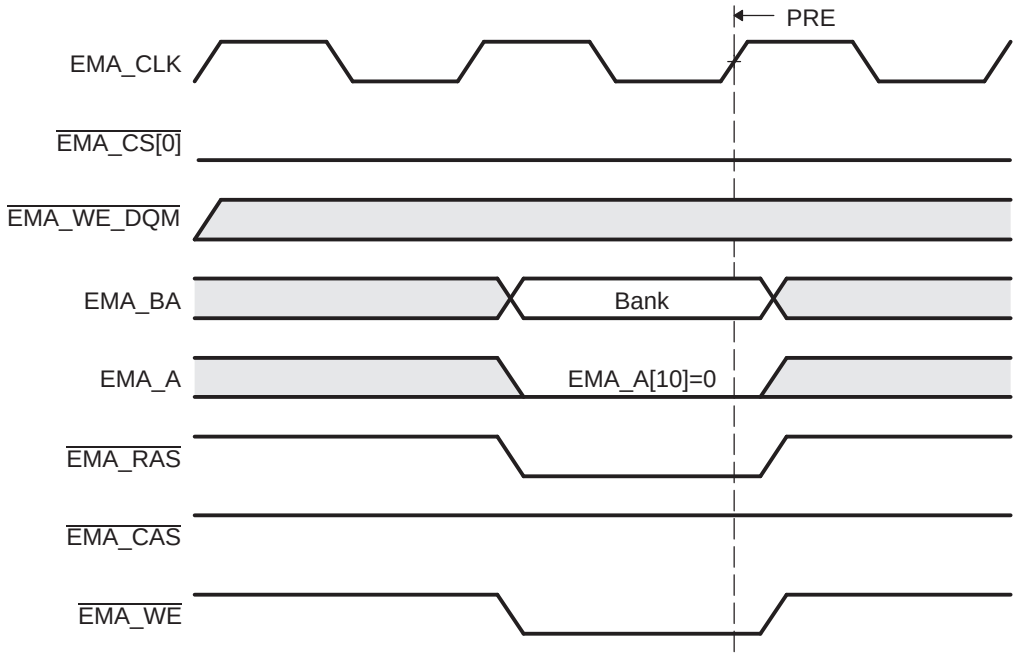
Table 4. EMIFA SDRAM Commands

Command	Function
PRE	**Precharge.** Depending on the value of EMA_A[10], the PRE command either deactivates the open row in all banks (EMA_A[10] = 1) or only the bank specified by the EMA_BA[1:0] pins (EMA_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 EMA_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 EMA_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 2.4.4.
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 5. 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]
EMIFA Pins:	EMA_SDCKE	$\overline{EMA\_CS[0]}$	$\overline{EMA\_RAS}$	$\overline{EMA\_CAS}$	$\overline{EMA\_WE}$	EMA_BA[1:0]	EMA_A[12:11]	EMA_A[10]	EMA_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



2.4.2 Interfacing to SDRAM

The EMIFA 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
- The number of internal banks is 1, 2, or 4

Figure 3 shows an interface between the EMIFA and a 2M x 16 x 4 bank SDRAM device, and Figure 4 shows an interface between the EMIFA and a 512K × 16 × 2 bank SDRAM device. For devices supporting 16-bit interface, refer to Table 6 for list of commonly-supported SDRAM devices and the required connections for the address pins.

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

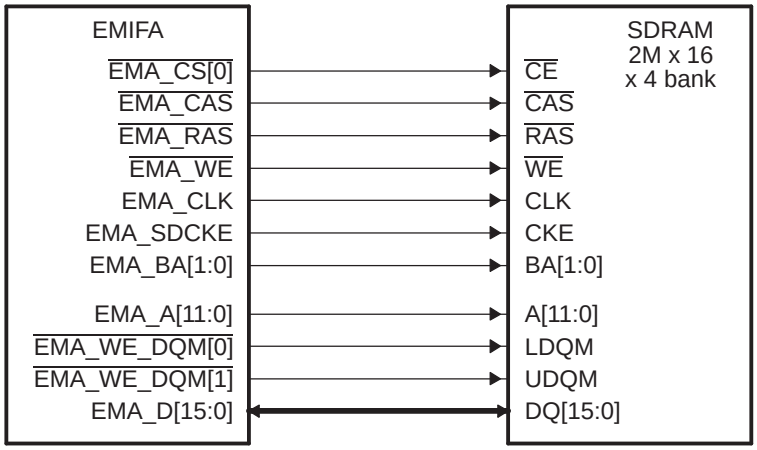


Figure 4. EMIFA to 512K × 16 × 2 bank SDRAM Interface

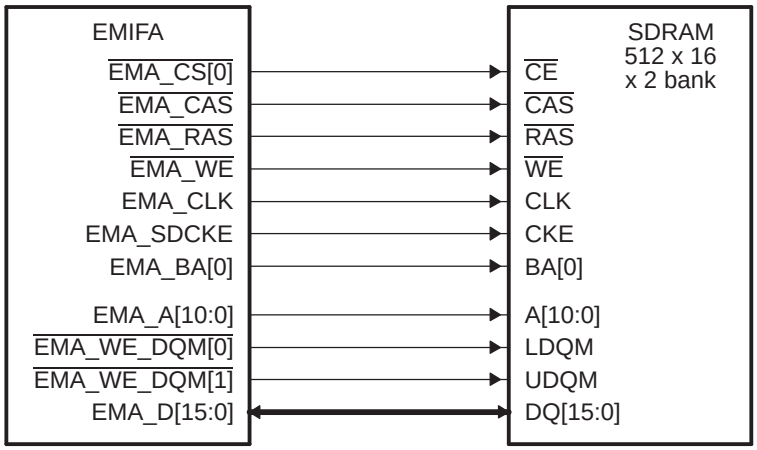


Table 6. 16-bit EMIFA Address Pin Connections

SDRAM Size	Width	Banks	Device	Address Pins
16M bits	×16	2	SDRAM	A[10:0]
			EMIFA	EMA_A[10:0]
64M bits	×16	4	SDRAM	A[11:0]
			EMIFA	EMA_A[11:0]
128M bits	×16	4	SDRAM	A[11:0]
			EMIFA	EMA_A[11:0]
256M bits	x16	4	SDRAM	A[12:0]
			EMIFA	EMA_A[12:0]
512M bits	x16	4	SDRAM	A[12:0]
			EMIFA	EMA_A[12:0]

2.4.3 SDRAM Configuration Registers

The operation of the EMIFA's SDRAM interface is controlled by programming the appropriate configuration registers. This section describes the purpose and function of each configuration register, but Section 3 should be referred for a more detailed description of each register, including the default registers values and bit-field positions. The following tables list the four such configuration registers, along with a description of each of their programmable fields.

NOTE: Writing to any of the fields: NM, CL, IBANK, and PAGESIZE in the SDRAM configuration register (SDCR) causes the EMIFA to abandon whatever it is currently doing and trigger the SDRAM initialization procedure described in Section 2.4.4.

Table 7. Description of the SDRAM Configuration Register (SDCR)

Parameter	Description
SR	This bit controls entering and exiting of the Self-Refresh mode. The field should be written using a byte-write to the upper byte of SDCR to avoid triggering the SDRAM initialization sequence.
PD	This bit controls entering and exiting of the Power down mode. The field should be written using a byte-write to the upper byte of SDCR to avoid triggering the SDRAM initialization sequence. If both SR and PD bits are set, the EMIFA will go into Self Refresh.
PDWR	Perform refreshes during Power Down. Writing a 1 to this bit will cause the EMIFA to exit the power down state and issue an AUTO REFRESH command every time Refresh May level is set. The field should be written using a byte-write to the upper byte of SDCR to avoid triggering the SDRAM initialization sequence. This bit should be set along with PD when entering power-down mode.
NM	**Narrow Mode.** This bit defines the width of the data bus between the EMIFA and the attached SDRAM device. When set to 1, the data bus is set to 16-bits. When set to 0, the data bus is set to 32-bits. This bit must always be set to 1.
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 2.4.4. Only, values of 2h (CAS latency = 2) and 3h (CAS latency = 3) are supported and should be written to this field. A 1 must be simultaneously written to the BIT11_9LOCK bit field of SDCR in order to write to the CL bit 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 2.4.11 for details.
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 • When PAGESIZE = 1h, 512-word pages are used • When PAGESIZE = 2h, 1024-word pages are used • When PAGESIZE = 3h, 2048-word pages are used This field value affects the mapping of logical addresses to SDRAM row, column, and bank addresses. See Section 2.4.11 for details.

Table 8. Description of the SDRAM Refresh Control Register (SDRCR)

Parameter	Description
RR	**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 RR for an SDRAM device: • RR = f_{EMA_CLK} / (Required SDRAM Refresh Rate) More information about the operation of the SDRAM refresh controller can be found in Section 2.4.6.

Table 9. Description of the SDRAM Timing Register (SDTIMR)

Parameter	Description
T_RFC T_RP T_RCD T_WR T_RAS T_RC T_RRD	**SDRAM Timing Parameters.** These fields configure the EMIFA to comply with the AC timing requirements of the attached SDRAM devices. This allows the EMIFA 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 3.6. These parameters should be set to satisfy the corresponding timing requirements found in the SDRAM's datasheet.

Table 10. Description of the SDRAM Self Refresh Exit Timing Register (SDSRETR)

Parameter	Description
T_XS	**Self Refresh Exit Parameter.** The T_XS field of this register informs the EMIFA about the minimum number of EMA_CLK cycles required between exiting Self Refresh and issuing any command. This parameter should be set to satisfy the t_{XSR} value for the attached SDRAM device.

2.4.4 SDRAM Auto-Initialization Sequence

The EMIFA automatically performs an SDRAM initialization sequence, regardless of whether it is interfaced to an SDRAM device, when either of the following two events occur:

- The EMIFA comes out of reset. No memory accesses to the SDRAM and Asynchronous interfaces are performed until this auto-initialization is complete.
- A write is performed to any of the three least significant bytes of the SDRAM configuration register (SDCR)

An SDRAM initialization sequence consists of the following steps:

1. If the initialization sequence is activated by a write to SDCR, and if any of the SDRAM banks are open, the EMIFA issues a PRE command with EMA_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.
2. The EMIFA drives EMA_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 RR field of SDRAM refresh control register (SDRCR), divided by the frequency of EMA_CLK (RR/f_{EMA_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 EMA_CLK, this step may or may not be sufficient to avoid violating the SDRAM constraint. See Section 2.4.5 for more information.
3. After the refresh intervals have elapsed, the EMIFA issues a PRE command with EMA_A[10] held high to indicate all banks.
4. The EMIFA issues eight AUTO REFRESH commands.
5. The EMIFA issues the LMR command with the EMA_A[9:0] pins set as described in Table 11.
6. Finally, the EMIFA performs a refresh cycle, which consists of the following steps:

   (a) Issuing a PRE command with EMA_A[10] held high if any banks are open
   (b) Issuing an REF command

Table 11. SDRAM LOAD MODE REGISTER Command

EMA_A[9:7]	EMA_A[6:4]	EMA_A[3]	EMA_A[2:0]
0 (Write bursts are of the programmed burst length in EMA_A[2:0])	These bits control the CAS latency of the SDRAM and are set according to CL field in the SDRAM configuration register (SDCR) as follows: • If CL = 2, EMA_A[6:4] = 2h (CAS latency = 2) • If CL = 3, EMA_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 (SDCR) as follows: • If NM = 0, EMA_A[2:0] = 2h (Burst Length = 4) • If NM = 1, EMA_A[2:0] = 3h (Burst Length = 8)

2.4.5 SDRAM Configuration Procedure

There are two different SDRAM configuration procedures. Although EMIFA automatically performs the SDRAM initialization sequence described in Section 2.4.4 when coming out of reset, it is recommended to follow one of the procedures listed below before performing any EMIFA memory requests. Procedure A should be followed if it is determined that the SDRAM Power-up constraint was not violated during the SDRAM Auto-Initialization Sequence detailed in Section 2.4.4 on coming out of Reset. The SDRAM Power-up constraint specifies that 200 μs (sometimes 100 μs) should exits between receiving stable Vdd and CLK and the issuing of a PRE command. Procedure B should be followed if the SDRAM Power-up constraint was violated. The 200 μs (100 μs) SDRAM Power-up constraint will be violated if the frequency of EMA_CLK is greater than 50 MHz (100 MHz for 100 μs SDRAM power-up constraint) during SDRAM Auto-Initialization Sequence. Procedure B should be followed if there is any doubt that the Power-up constraint was met.

Following is the procedure to be followed if the SDRAM Power-up constraint was NOT violated (Procedure A):

1. Place the SDRAM into Self-Refresh Mode by setting the SR bit of SDCR to 1. A byte-write to the upper byte of SDCR should be used to avoid restarting the SDRAM Auto-Initialization Sequence described in Section 2.4.4. The SDRAM should be placed into Self-Refresh mode when changing the frequency of EMA_CLK to avoid incurring the 200 μs Power-up constraint again.
2. Program the CPU's PLL Controller to provide the desired EMA_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. Remove the SDRAM from Self-Refresh Mode by clearing the SR bit of SDCR to 0. A byte-write to the upper byte of SDCR should be used to avoid restarting the SDRAM Auto-Initialization Sequence described in Section 2.4.4.
4. Program SDTIMR and SDSRETR to satisfy the timing requirements for the attached SDRAM device. The timing parameters should be taken from the SDRAM datasheet.
5. Program the RR field of SDRCR to match that of the attached device's refresh interval. See Section 2.4.6.1 details on determining the appropriate value.
6. Program SDCR to match the characteristics of the attached SDRAM device. This will cause the auto-initialization sequence in Section 2.4.4 to be re-run. This second initialization generally takes much less time due to the increased frequency of EMA_CLK.

Following is the procedure to be followed if the SDRAM Power-up constraint was violated (Procedure B):

1. Program the CPU's PLL Controller to provide the desired EMA_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.
2. Program SDTIMR and SDSRETR to satisfy the timing requirements for the attached SDRAM device. The timing parameters should be taken from the SDRAM datasheet.
3. Program the RR field of SDRCR such that the following equation is satisfied: $(RR \times 8)/(f_{EMA\_CLK}) >$ 200 μs (sometimes 100 μs). For example, an EMA_CLK frequency of 100 MHz would require setting RR to 2501 (9C5h) or higher to meet a 200 μs constraint.

4. Program SDCR to match the characteristics of the attached SDRAM device. This will cause the auto-initialization sequence in Section 2.4.4 to be re-run with the new value of RR.
5. Perform a read from the SDRAM to assure that step 5 of this procedure will occur after the initialization process has completed. Alternatively, wait for 200 μs instead of performing a read.
6. Finally, program the RR field to match that of the attached device's refresh interval. See Section 2.4.6.1 details on determining the appropriate value.

After following the above procedure, the EMIFA is ready to perform accesses to the attached SDRAM device. See Appendix A for an example of configuring the SDRAM interface.

2.4.6 EMIFA Refresh Controller

An SDRAM device requires that each of its rows be refreshed at a minimum required rate. The EMIFA 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 EMIFA of the required rate for performing auto refresh cycles, the RR field of the SDRAM refresh control register (SDRCR) must be programmed. The EMIFA 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 EMIFA is not interfaced with an SDRAM. The remainder of this section details the EMIFA's refresh scheme and provides an example for determining the appropriate value to place in the RR field of SDRCR.

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

- The refresh interval counter is reloaded with the value from the RR 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 EMIFA 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 EMIFA uses the refresh backlog counter to determine the urgency with which an auto refresh cycle should be performed. The four levels of urgency are described in Table 12. This refresh scheme allows the required refreshes to be performed with minimal impact on access requests.

Table 12. Refresh Urgency Levels

Urgency Level	Refresh Backlog Counter Range	Action Taken
Refresh May	1-3	An auto-refresh cycle is performed only if the EMIFA has no requests pending and none of the SDRAM banks are open.
Refresh Release	4-7	An auto-refresh cycle is performed if the EMIFA 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 EMIFA can begin servicing any new read or write requests.

2.4.6.1 Determining the Appropriate Value for the RR Field

The value that should be programmed into the RR field of SDRCR can be calculated by using the frequency of the EMA_CLK signal (f_{EMA_CLK}) and the required refresh rate of the SDRAM ($f_{Refresh}$). The following formula can be used:

$RR = f_{EMA\_CLK} / f_{Refresh}$

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 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 that should be programmed into the RR field can be computed as:

$RR = f_{EMA\_CLK} \times t_{Refresh\ Period} / n_{cycles}$

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

- f_{EMA_CLK} = 100 MHz (frequency of the EMIFA 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)

RR can be calculated as:

RR = 100 MHz × 64 ms/8192

RR = 781.25

RR = 782 cycles = 30Eh cycles

2.4.7 Self-Refresh Mode

The EMIFA can be programmed to enter the self-refresh state by setting the SR bit of SDCR to 1. This will cause the EMIFA 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. The SR bit should be set and cleared using a byte-write to the upper byte of the SDRAM configuration register (SDCR) to avoid triggering the SDRAM initialization sequence.

While in the self-refresh state, the EMIFA continues to service asynchronous bank requests and register accesses as normal, with one caveat. The EMIFA will not park the data bus following a read to asynchronous memory while in the self-refresh state. Instead, the EMIFA tri-states the data bus. Therefore, it is not recommended to perform asynchronous read operations while the EMIFA is in the self-refresh state, in order to prevent floating inputs on the data bus. More information about data bus parking can be found in Section 2.6.

The EMIFA will exit from the self-refresh state if either of the following events occur:

- The SR bit of SDCR is cleared to 0.
- An SDRAM accesses is requested.

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

The attached SDRAM device should also be placed into Self-Refresh Mode when changing the frequency of EMA_CLK using the PLL Controller. If the frequency of EMA_CLK changes while the SDRAM is not in Self-Refresh Mode, Procedure B in Section 2.4.5 should be followed to reinitialize the device.

2.4.8 Power Down Mode

To support low-power modes, the EMIFA can be requested to issue a POWER DOWN command to the SDRAM by setting the PD bit in the SDRAM configuration register (SDCR). When this bit is set, the EMIFA 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 EMIFA will enter the power-down state. Upon entering this state, the EMIFA will issue a POWER DOWN command (same as a NOP command but driving EMA_SDCKE low on the same cycle). The EMIFA then maintains EMA_SDCKE low until it exits the power-down state.

Since the EMIFA services the refresh backlog before it enters the power-down state, all internal banks of the SDRAM are closed (precharged) prior to issuing the POWER DOWN command. Therefore, the EMIFA only supports Precharge Power Down. The EMIFA does not support Active Power Down, where internal banks of the SDRAM are open (active) before the POWER DOWN command is issued.

During the power-down state, the EMIFA services the SDRAM, asynchronous memory, and register accesses as normal, returning to the power-down state upon completion.

The PDWR bit in SDCR indicates whether the EMIFA should perform refreshes in power-down state. If the PDWR bit is set, the EMIFA exits the power-down state every time the Refresh Must level is set, performs AUTO REFRESH commands to the SDRAM, and returns back to the power-down state. This evenly distributes the refreshes to the SDRAM in power-down state. If the PDWR bit is not set, the EMIFA does not perform any refreshes to the SDRAM. Therefore, the data integrity of the SDRAM is not assured upon power down exit if the PDWR bit is not set.

If the PD bit is cleared while in the power-down state, the EMIFA will come out of the power-down state. The EMIFA:

- Drives EMA_SDCKE high.
- Enters its idle state.

2.4.9 SDRAM Read Operation

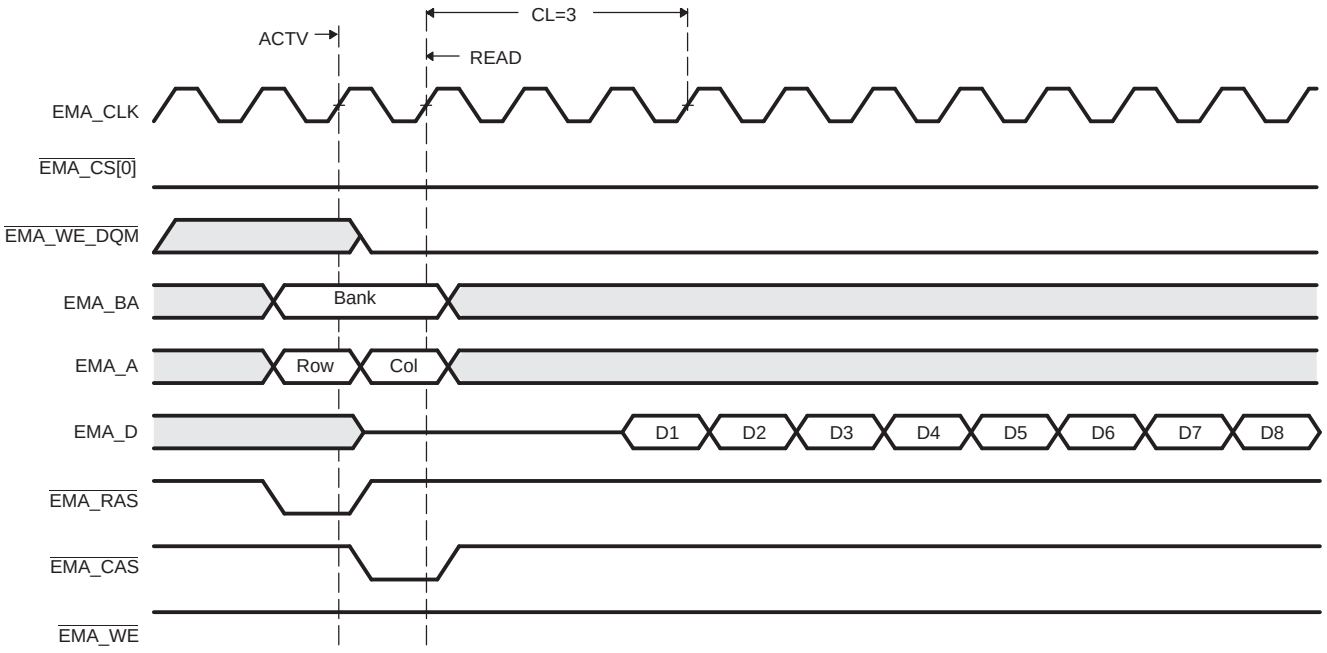
When the EMIFA receives a read request to SDRAM from one of the requesters listed in Section 2.2, it performs one or more read access cycles. A read access cycle begins with the issuing of the ACTV command to select the desired bank and row of the SDRAM device. After the row has been opened, the EMIFA proceeds to issue a READ command while specifying the desired bank and column address. EMA_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 EMIFA issues NOP commands. Following a READ command, the CL field of the SDRAM configuration register (SDCR) 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 5 shows the signal waveforms for a basic SDRAM read operation in which a burst of data is read from a single page. When the EMIFA SDRAM interface is configured to 16 bit by setting the NM bit of the SDRAM configuration register (SDCR) to 1, a burst size of eight is used. Figure 5 shows a burst size of eight.

The EMIFA will truncate a series of bursting data if the remaining addresses of the burst are not required to complete the request. The EMIFA 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.

Figure 5. Timing Waveform for Basic SDRAM Read Operation



Several other pins are also active during a read access. The EMA_WE_DQM[1:0] 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 EMIFA pins during each command can be found in Table 5.

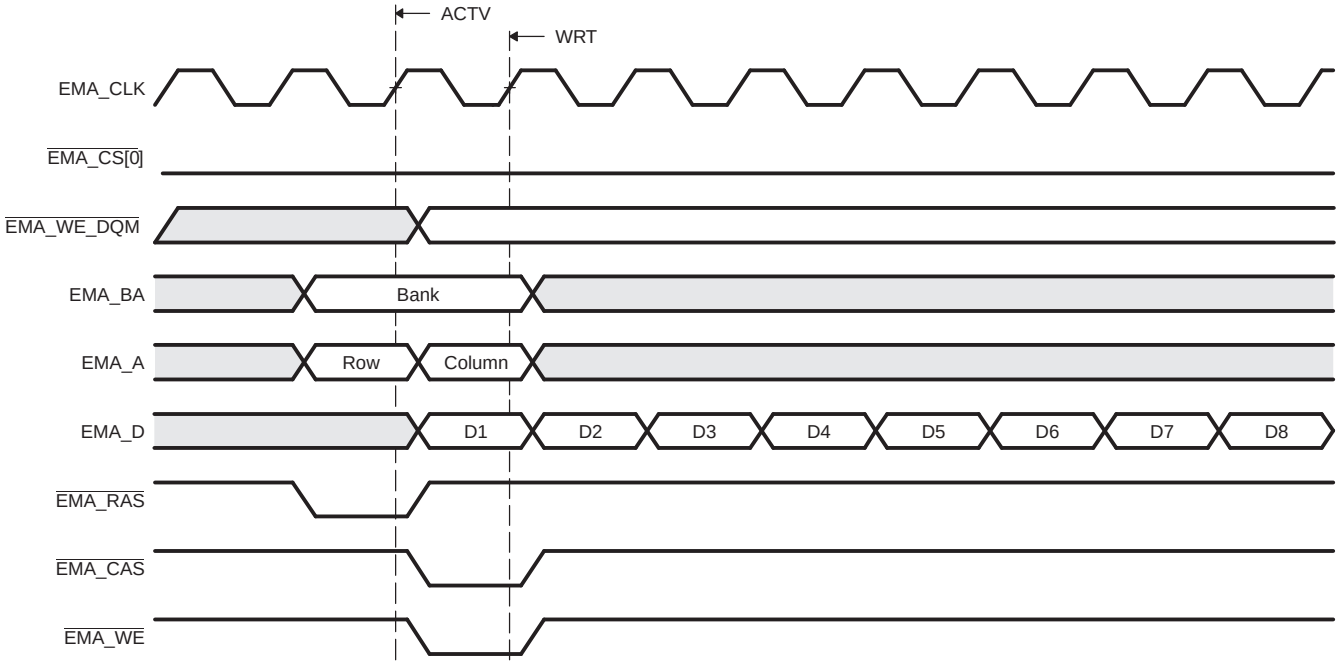
The EMIFA schedules its commands based on the timing information that is provided to it in the SDRAM timing register (SDTIMR). The values for the timing parameters in this register should be chosen to satisfy the timing requirements listed in the SDRAM datasheet. The EMIFA uses this timing information 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 SDTIMR in Section 3.6 for more details on the various timing parameters.

2.4.10 SDRAM Write Operations

When the EMIFA receives a write request to SDRAM from one of the requesters listed in Section 2.2, 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 EMIFA proceeds to issue a WRT command while specifying the desired bank and column address. EMA_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 EMIFA 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 6 shows the signal waveforms for a basic SDRAM write operation in which a burst of data is read from a single page. When the EMIFA SDRAM interface is configured to 16-bit by setting the NM bit of the SDRAM configuration register (SDCR) to 1, a burst size of eight is used. Figure 6 shows a burst size of eight.

Figure 6. Timing Waveform for Basic SDRAM Write Operation



The EMIFA will truncate a series of bursting data if the remaining addresses of the burst are not part of the write request. The EMIFA 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 EMA_WE_DQM[1:0] 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 EMIFA pins during each command can be found in Table 5.

The EMIFA schedules its commands based on the timing information that is provided to it in the SDRAM timing register (SDTIMR). The values for the timing parameters in this register should be chosen to satisfy the timing requirements listed in the SDRAM datasheet. The EMIFA 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 SDTIMR in Section 3.6 for more details on the various timing parameters.

2.4.11 Mapping from Logical Address to EMIFA Pins

When the EMIFA 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 this address mapping are shown in Table 13 for 16-bit operation. Using the settings of the IBANK and PAGESIZE fields of the SDRAM configuration register (SDCR), the EMIFA 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 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 EMIFA moves into the same page in the next bank of the attached device by incrementing the bank address EMA_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 that can be open at one time, but only one page within a bank can be open at a time.

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

Table 13. Mapping from Logical Address to EMIFA Pins for 16-bit SDRAM

IBANK	PAGESIZE	Logical Address													
		31:27	26	25	24	23	22	21:14	13	12	11	10	9	8:1	0
0	0	-						Row Address						Col Address	$\overline{\text{EMA\_WE\_DQM}}$[0]
1	0	-						Row Address					EMA_BA[0]	Col Address	$\overline{\text{EMA\_WE\_DQM}}$[0]
2	0	-						Row Address				EMA_BA[1:0]		Col Address	$\overline{\text{EMA\_WE\_DQM}}$[0]
0	1	-						Row Address					Column Address		$\overline{\text{EMA\_WE\_DQM}}$[0]
1	1	-						Row Address				EMA_BA[0]	Column Address		$\overline{\text{EMA\_WE\_DQM}}$[0]
2	1	-						Row Address			EMA_BA[1:0]		Column Address		$\overline{\text{EMA\_WE\_DQM}}$[0]
0	2	-						Row Address				Column Address			$\overline{\text{EMA\_WE\_DQM}}$[0]
1	2	-						Row Address			EMA_BA[0]	Column Address			$\overline{\text{EMA\_WE\_DQM}}$[0]
2	2	-						Row Address		EMA_BA[1:0]		Column Address			$\overline{\text{EMA\_WE\_DQM}}$[0]
0	3	-						Row Address			Column Address				$\overline{\text{EMA\_WE\_DQM}}$[0]
1	3	-						Row Address		EMA_BA[0]	Column Address				$\overline{\text{EMA\_WE\_DQM}}$[0]
2	3	-						Row Address		EMA_BA[1:0]	Column Address				$\overline{\text{EMA\_WE\_DQM}}$[0]

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

2.5 Asynchronous Controller and Interface

The EMIFA easily interfaces to a variety of asynchronous devices including NOR Flash, NAND Flash, and SRAM. It can be operated in two major modes (see Table 14):

- Normal Mode
- Select Strobe Mode

Table 14. Normal Mode vs. Select Strobe Mode

Mode	Function of $\overline{\text{EMA\_WE\_DQM}}$ pins	Operation of $\overline{\text{EMA\_CS[5:2]}}$
Normal Mode	Byte enables	Active during the entire asynchronous access cycle
Select Strobe Mode	Byte enables	Active only during the strobe period of an access cycle

The first mode of operation is Normal Mode, in which the $\overline{\text{EMA\_WE\_DQM}}$ pins of the EMIFA function as byte enables. In this mode, the $\overline{\text{EMA\_CS[5:2]}}$ pins behaves as typical chip select signals, remaining active for the duration of the asynchronous access. See Section 2.5.1 for an example interface with multiple 8-bit devices.

The second mode of operation is Select Strobe Mode, in which the $\overline{\text{EMA\_CS[5:2]}}$ pins act as a strobe, active only during the strobe period of an access. In this mode, the $\overline{\text{EMA\_WE\_DQM}}$ pins of the EMIFA function as standard byte enables for reads and writes. A summary of the differences between the two modes of operation are shown in Table 14. Refer to Section 2.5.4 for the details of asynchronous operations in Normal Mode, and to Section 2.5.5 for the details of asynchronous operations in Select Strobe Mode. The EMIFA hardware defaults to Normal Mode, but can be manually switched to Select Strobe Mode by setting the SS bit in the asynchronous *m* (m = 1, 2, 3, or 4) configuration register (CE*n*CFG) (*n* = 2, 3, 4, or 5). Throughout the document *m* can hold the values 1, 2, 3 or 4; and *n* can hold the values 2, 3, 4, or 5.

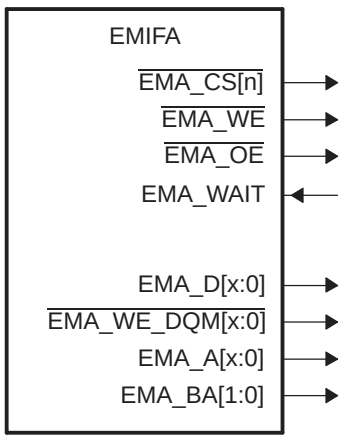
In both Normal Mode and Select Strobe Mode, the EMIFA can be configured to operate in a sub-mode called NAND Flash Mode. In NAND Flash Mode, the EMIFA is able to calculate an error correction code (ECC) for transfers up to 512 bytes.

The EMIFA also provides configurable cycle timing parameters and an Extended Wait Mode that allows the connected device to extend the strobe period of an access cycle. The following sections describe the features related to interfacing with external asynchronous devices.

2.5.1 Interfacing to Asynchronous Memory

Figure 7 shows the EMIFA's external pins used in interfacing with an asynchronous device. In $\overline{\text{EMA\_CS[n]}}$, n = 2, 3, 4, or 5.

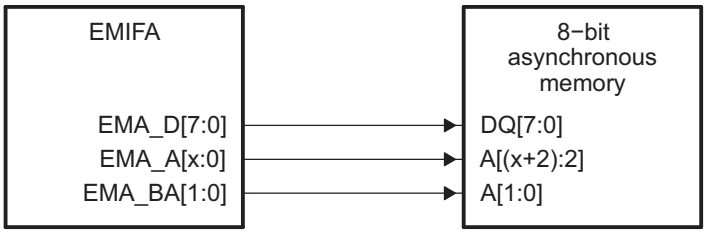
Figure 7. EMIFA Asynchronous Interface



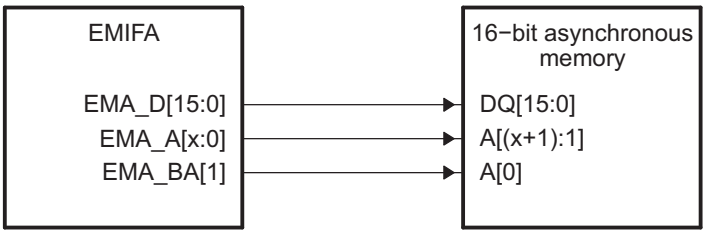
Of special note is the connection between the EMIFA and the external device's address bus. The EMIFA address pin EM_A[0] always provides the least significant bit of a 32-bit word address. Therefore, when interfacing to a 16-bit or 8-bit asynchronous device, the EMA_BA[1] and EMA_BA[0] pins provide the least-significant bits of the halfword or byte address, respectively. Additionally, when the EMIFA interfaces to a 16-bit asynchronous device, the EMA_BA[0] pin can serve as the upper address line EM_A[22]. Note that the width of the address bus varies with devices; therefore, see your device-specific data manual for the EM_A bus width supported. Figure 8 and Figure 9 show the mapping between the EMIFA and the connected device's data and address pins for various programmed data bus widths. The data bus width may be configured in the asynchronous *n* configuration register (CE*n*CFG).

Figure 9 shows a common interface between the EMIFA and external asynchronous memory. Figure 9 shows an interface between the EMIFA and an external memory with byte enables. The EMIFA should be operated in either Normal Mode or Select Strobe Mode when using this interface, so that the $\overline{\text{EMA\_WE\_DQM}}$ signals operate as byte enables.

Figure 8. EMIFA to 8-bit/16-bit Memory Interface

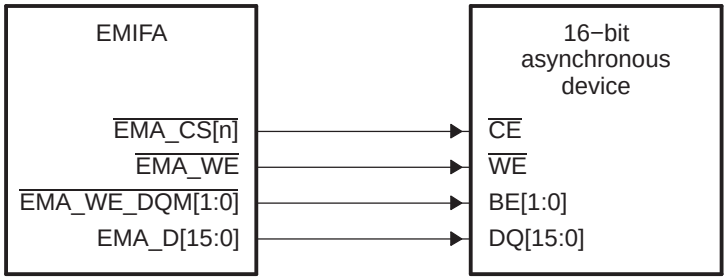


a) EMIF to 8-bit memory interface



b) EMIF to 16-bit memory interface

Figure 9. Common Asynchronous Interface



2.5.2 Accessing Larger Asynchronous Memories

The device has a limited number of dedicated EMIFA address pins, enough to interface directly to an SDRAM. If a device such as an asynchronous flash needs to be attached to the EMIFA, then GPIO pins may be used to control the flash device's upper address lines. This is sufficient to boot from the flash. Normally, code stored in flash is copied into SDRAM or internal memory before executing because these memories have much faster access times. For details on which device pins are GPIO capable, see your device-specific data manual.

The ROM bootloader can load a secondary bootloader from an attached asynchronous device. The ROM bootloader assumes that any GPIO pins used to control the upper address lines of the boot flash will be pulled to 0 after reset. This means that normally the GPIO pins selected for this function will be either spare or used as outputs only by the application, and therefore can be pulled to 0 at reset with an external pulldown resistor. The GPIO pins chosen should be tri-stated by default on device reset. For details on which GPIO-capable pins are tri-stated on device reset, see your device-specific data manual.

When booting from flash, the ROM bootloader copies a board-specific secondary bootloader from the lower portion of the flash, so it does not need to manipulate the upper address lines. Only the secondary bootloader, which is board-specific and is stored in the external flash, needs to know which GPIO pins have been assigned to the function of upper address lines. Therefore, the secondary bootloader can perform the task of configuring the selected pins as GPIO and loading the remainder of the code from the upper flash memory.

2.5.3 Configuring the EMIFA for Asynchronous Accesses

The operation of the EMIFA's asynchronous interface can be configured by programming the appropriate register fields. The reset value and bit position for each register field can be found in Section 3, but the Boot ROM documentation should be consulted to determine if the fields are programmed during boot. The following tables list the register fields that can be programmed and describe the purpose of each field. These registers can be programmed prior to accessing the external memory, and the transfer following a write to these registers will use the new configuration.

Table 15. Description of the Asynchronous *m* Configuration Register (CE*n*CFG)

Parameter	Description
SS	**Select Strobe mode.** This bit selects the EMIFA's mode of operation in the following way: • SS = 0 selects Normal Mode – $\overline{\text{EMA\_WE\_DQM}}$ pins function as byte enables – $\overline{\text{EMA\_CS}}$[5:2] active for duration of access • SS = 1 selects Select Strobe Mode – $\overline{\text{EMA\_WE\_DQM}}$ pins function as byte enables – $\overline{\text{EMA\_CS}}$[5:2] acts as a strobe.
EW	**Extended Wait Mode enable.** • EW = 0 disables Extended Wait Mode • EW = 1 enables Extended Wait Mode When set to 1, the EMIFA enables its Extended Wait Mode in which the strobe width of an access cycle can be extended in response to the assertion of the EMA_WAIT pin(1). The WP*n* bit in the asynchronous wait cycle configuration register (AWCC) controls to polarity of EMA_WAIT pin. Extended Wait Mode should not be used while in NAND Flash Mode. See Section 2.5.7 for more details on this mode of operation.
W_SETUP/R_SETUP	**Read/Write setup widths.** These fields define the number of EMIFA clock cycles of setup time for the address pins (EMA_A and EMA_BA), byte enables ($\overline{\text{EMA\_WE\_DQM}}$), and asynchronous chip enable ($\overline{\text{EMA\_CS}}$[5:2]) before the read strobe pin ($\overline{\text{EMA\_OE}}$) or write strobe pin ($\overline{\text{EMA\_WE}}$) falls, minus one cycle. For writes, the W_SETUP field also defines the setup time for the data pins (EMA_D). Refer to the datasheet of the external asynchronous device to determine the appropriate setting for this field.
W_STROBE/R_STROBE	**Read/Write strobe widths.** These fields define the number of EMIFA clock cycles between the falling and rising of the read strobe pin ($\overline{\text{EMA\_OE}}$) or write strobe pin ($\overline{\text{EMA\_WE}}$), minus one cycle. If Extended Wait Mode is enabled by setting the EW field in the asynchronous *n* configuration register (CE*n*CFG), these fields must be set to a value greater than zero. Refer to the datasheet of the external asynchronous device to determine the appropriate setting for this field.

(1) The EMA_WAIT pin is not available on all devices; therefore, this field is reserved on those devices.

Table 15. Description of the Asynchronous *m* Configuration Register (CE*n*CFG) (continued)

Parameter	Description
W_HOLD/R_HOLD	**Read/Write hold widths.** These fields define the number of EMIFA clock cycles of hold time for the address pins (EMA_A and EMA_BA), byte enables (EMA_WE_DQM), and asynchronous chip enable (EMA_CS[5:2]) after the read strobe pin (EMA_OE) or write strobe pin (EMA_WE) rises, minus one cycle. For writes, the W_HOLD field also defines the hold time for the data pins (EMA_D). Refer to the datasheet of the external asynchronous device to determine the appropriate setting for this field.
TA	**Minimum turnaround time.** This field defines the minimum number of EMIFA clock cycles between asynchronous reads and writes, minus one cycle. The purpose of this feature is to avoid contention on the bus. The value written to this field also determines the number of cycles that will be inserted between asynchronous accesses and SDRAM accesses. Refer to the datasheet of the external asynchronous device to determine the appropriate setting for this field.
ASIZE	**Asynchronous Device Bus Width.** This field determines the data bus width of the asynchronous interface in the following way: • ASIZE = 0 selects an 8-bit bus • ASIZE = 1 selects a 16-bit bus The configuration of ASIZE determines the function of the EMA_A and EMA_BA pins as described in Section 2.5.1. This field also determines the number of external accesses required to fulfill a request generated by one of the sources mentioned in Section 2.2. For example, a request for a 32-bit word would require four external access when ASIZE = 0. Refer to the datasheet of the external asynchronous device to determine the appropriate setting for this field.

Table 16. Description of the Asynchronous Wait Cycle Configuration Register (AWCC)[(1)]

Parameter	Description
WP*n*	**EM_WAIT Polarity.** • WP*n* = 0 selects active-low polarity • WP*n* = 1 selects active-high polarity When set to 1, the EMIFA will wait if the EMA_WAIT pin is high. When cleared to 0, the EMIFA will wait if the EMA_WAIT pin is low. The EMIFA must have the Extended Wait Mode enabled for the EMA_WAIT pin to affect the width of the strobe period. The polarity of the EMA_WAIT signal is not programmable in NAND Flash Mode.
MAX_EXT_WAIT	**Maximum Extended Wait Cycles.** This field configures the number of EMIFA clock cycles the EMIFA will wait for the EMA_WAIT pin to be deactivated during the strobe period of an access cycle. The maximum number of EMIFA clock cycles it will wait is determined by the following formula: Maximum Extended Wait Cycles = (MAX_EXT_WAIT + 1) × 16 If the EMA_WAIT pin is not deactivated within the time specified by this field, the EMIFA resumes the access cycle, registering whatever data is on the bus and proceeding to the hold period of the access cycle. This situation is referred to as an Asynchronous Timeout. An Asynchronous Timeout generates an interrupt, if it has been enabled in the EMIFA interrupt mask set register (INTMSKSET). Refer to Section 2.8.1 for more information about the EMIFA interrupts. Extended Wait Mode should not be used while in NAND Flash Mode.

(1) The EMA_WAIT pin is not available on all devices; therefore, this register is reserved on those devices.

Table 17. Description of the EMIFA Interrupt Mask Set Register (INTMSKSET)

Parameter	Description
WR_MASK_SET	**Wait Rise Mask Set.** Writing a 1 enables an interrupt to be generated when a rising edge on EMA_WAIT[1] occurs while in NAND Flash Mode
AT_MASK_SET	**Asynchronous Timeout Mask Set.** Writing a 1 to this bit enables an interrupt to be generated when an Asynchronous Timeout occurs.

(1) The EMA_WAIT pin is not available on all devices; therefore, this field is reserved on those devices.

Table 18. Description of the EMIFA Interrupt Mast Clear Register (INTMSKCLR)

Parameter	Description
WR_MASK_CLR	**Wait Rise Mask Clear.** Writing a 1 to this bit disables the interrupt, clearing the WR_MASK_SET bit in the EMIFA interrupt mask set register (INTMSKSET).
AT_MASK_CLR	**Asynchronous Timeout Mask Clear.** Writing a 1 to this bit prevents an interrupt from being generated when an Asynchronous Timeout occurs.

2.5.4 Read and Write Operations in Normal Mode

Normal Mode is the asynchronous interface's default mode of operation. It is selected when the SS bit in the asynchronous *n* configuration register (CE*n*CFG) is cleared to 0. In this mode, the EMA_WE_DQM pins operate as byte enables. Section 2.5.4.1 and Section 2.5.4.2 explain the details of read and write operations while in Normal Mode.

2.5.4.1 Asynchronous Read Operations (Normal Mode)

NOTE: During the entirety of an asynchronous read operation, the EMA_WE pin is driven high.

An asynchronous read is performed when any of the requesters mentioned in Section 2.2 request a read from the attached asynchronous memory. After the request is received, a read operation is initiated once it becomes the EMIFA's highest priority task, according to the priority scheme detailed in Section 2.12. In the event that the read request cannot be serviced by a single access cycle to the external device, multiple access cycles will be performed by the EMIFA until the entire request is fulfilled. The details of an asynchronous read operation in Normal Mode are described in Table 19. Also, Figure 10 shows an example timing diagram of a basic read operation.

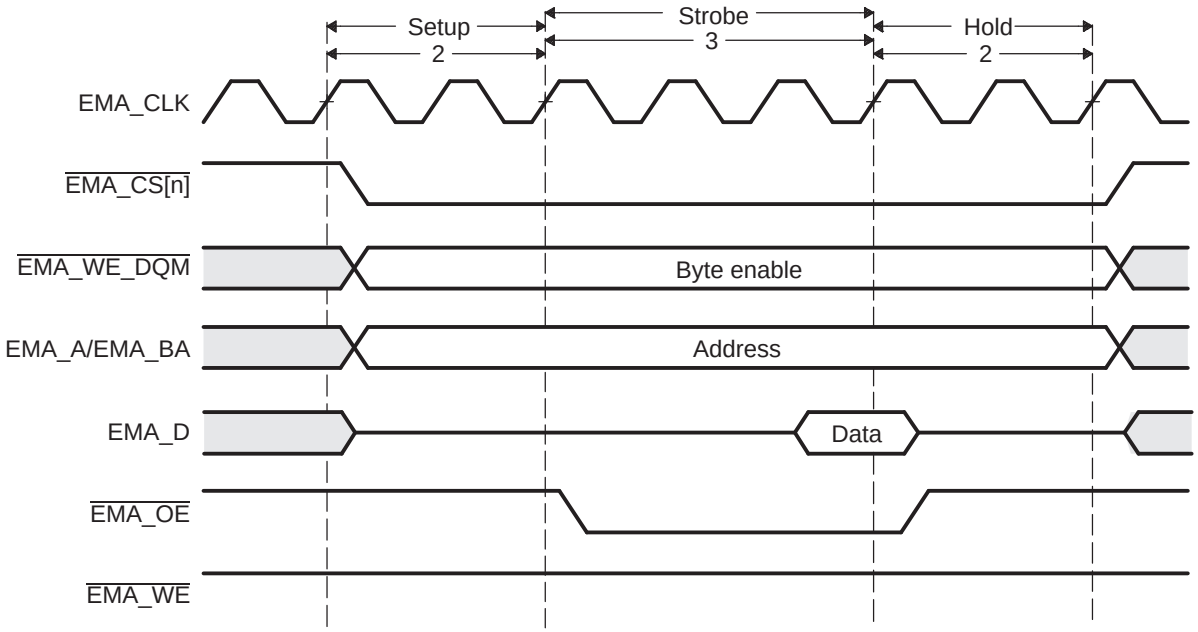
Table 19. Asynchronous Read Operation in Normal Mode

Time Interval	Pin Activity in Normal Mode
Turn-around period	Once the read operation becomes the highest priority task for the EMIFA, the EMIFA waits for the programmed number of turn-around cycles before proceeding to the setup period of the operation. The number of wait cycles is taken directly from the TA field of the asynchronous *n* configuration register (CE*n*CFG). There are two exceptions to this rule: • If the current read operation was directly proceeded by another read operation, no turnaround cycles are inserted. • If the current read operation was directly proceeded by a write operation and the TA field has been cleared to 0, one turn-around cycle will be inserted. After the EMIFA has waited for the turnaround cycles to complete, it again checks to make sure that the read operation is still its highest priority task. If so, the EMIFA proceeds to the setup period of the operation. If it is no longer the highest priority task, the EMIFA terminates the operation.

Table 19. Asynchronous Read Operation in Normal Mode (continued)

Time Interval	Pin Activity in Normal Mode
Start of the setup period	The following actions occur at the start of the setup period: • The setup, strobe, and hold values are set according to the R_SETUP, R_STROBE, and R_HOLD values in CE*n*CFG. • The address pins EMA_A and EMA_BA become valid and carry the values described in Section 2.5.1. • $\overline{\text{EMA\_CS}}$[5:2] falls to enable the external device (if not already low from a previous operation)
Strobe period	The following actions occur during the strobe period of a read operation: 1. $\overline{\text{EMA\_OE}}$ falls at the start of the strobe period 2. On the rising edge of the clock which is concurrent with the end of the strobe period: • $\overline{\text{EMA\_OE}}$ rises • The data on the EMA_D bus is sampled by the EMIFA. In Figure 10, EMA_WAIT is inactive. If EMA_WAIT is instead activated, the strobe period can be extended by the external device to give it more time to provide the data. Section 2.5.7 contains more details on using the EMA_WAIT pin.
End of the hold period	At the end of the hold period: • The address pins EMA_A and EMA_BA become invalid • $\overline{\text{EMA\_CS}}$[5:2] rises (if no more operations are required to complete the current request) EMIFA may be required to issue additional read operations to a device with a small data bus width in order to complete an entire word access. In this case, the EMIFA immediately re-enters the setup period to begin another operation without incurring the turn-round cycle delay. The setup, strobe, and hold values are not updated in this case. If the entire word access has been completed, the EMIFA returns to its previous state unless another asynchronous request has been submitted and is currently the highest priority task. If this is the case, the EMIFA instead enters directly into the turnaround period for the pending read or write operation.

Figure 10. Timing Waveform of an Asynchronous Read Cycle in Normal Mode



2.5.4.2 Asynchronous Write Operations (Normal Mode)

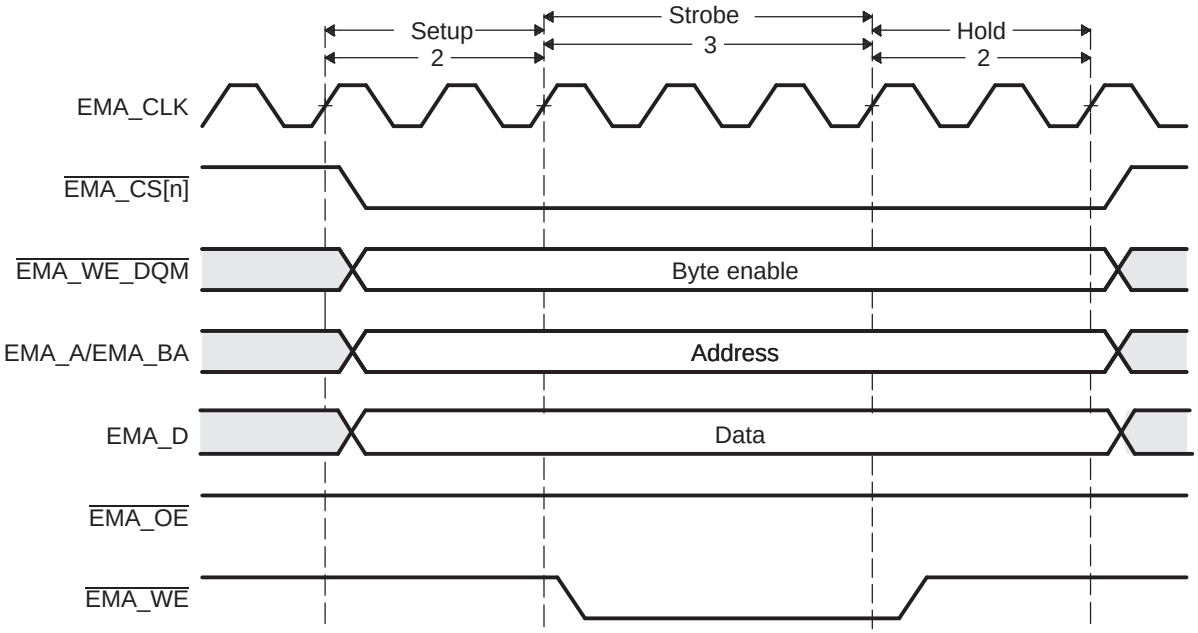
NOTE: During the entirety of an asynchronous write operation, the EMA_OE pin is driven high.

An asynchronous write is performed when any of the requesters mentioned in Section 2.2 request a write to memory in the asynchronous bank of the EMIFA. After the request is received, a write operation is initiated once it becomes the EMIFA's highest priority task, according to the priority scheme detailed in Section 2.12. In the event that the write request cannot be serviced by a single access cycle to the external device, multiple access cycles will be performed by the EMIFA until the entire request is fulfilled. The details of an asynchronous write operation in Normal Mode are described in Table 20. Also, Figure 11 shows an example timing diagram of a basic write operation.

Table 20. Asynchronous Write Operation in Normal Mode

Time Interval	Pin Activity in Normal Mode
Turnaround period	Once the write operation becomes the highest priority task for the EMIFA, the EMIFA waits for the programmed number of turn-around cycles before proceeding to the setup period of the operation. The number of wait cycles is taken directly from the TA field of the asynchronous *n* configuration register (CE*n*CFG). There are two exceptions to this rule: • If the current write operation was directly proceeded by another write operation, no turn-around cycles are inserted. • If the current write operation was directly proceeded by a read operation and the TA field has been cleared to 0, one turnaround cycle will be inserted. After the EMIFA has waited for the turn-around cycles to complete, it again checks to make sure that the write operation is still its highest priority task. If so, the EMIFA proceeds to the setup period of the operation. If it is no longer the highest priority task, the EMIFA terminates the operation.
Start of the setup period	The following actions occur at the start of the setup period: • The setup, strobe, and hold values are set according to the W_SETUP, W_STROBE, and W_HOLD values in CE*n*CFG. • The address pins EMA_A and EMA_BA and the data pins EMA_D become valid. The EMA_A and EMA_BA pins carry the values described in Section 2.5.1. • EMA_CS[5:2] falls to enable the external device (if not already low from a previous operation).
Strobe period	The following actions occur at the start of the strobe period of a write operation: 1. EMA_WE falls 2. The EMA_WE_DQM pins become valid as byte enables. The following actions occur on the rising edge of the clock which is concurrent with the end of the strobe period: 1. EMA_WE rises 2. The EMA_WE_DQM pins deactivate In Figure 11, EMA_WAIT is inactive. If EMA_WAIT is instead activated, the strobe period can be extended by the external device to give it more time to accept the data. Section 2.5.7 contains more details on using the EMA_WAIT pin.
End of the hold period	At the end of the hold period: • The address pins EMA_A and EMA_BA become invalid • The data pins become invalid • EMA_CS[n] (n = 2,3,4, or 5) rises (if no more operations are required to complete the current request) The EMIFA may be required to issue additional write operations to a device with a small data bus width in order to complete an entire word access. In this case, the EMIFA immediately re-enters the setup period to begin another operation without incurring the turnaround cycle delay. The setup, strobe, and hold values are not updated in this case. If the entire word access has been completed, the EMIFA returns to its previous state unless another asynchronous request has been submitted and is currently the highest priority task. If this is the case, the EMIFA instead enters directly into the turnaround period for the pending read or write operation.

Figure 11. Timing Waveform of an Asynchronous Write Cycle in Normal Mode



2.5.5 Read and Write Operation in Select Strobe Mode

Select Strobe Mode is the EMIFA's second mode of operation. It is selected when the SS bit of the asynchronous *n* configuration register (CE*n*CFG) is set to 1. In this mode, the EMA_WE_DQM pins operate as byte enables and the EMA_CS[n] (n = 2,3,4, or 5) pin is only active during the strobe period of an access cycle. Section 2.5.4.1 and Section 2.5.4.2 explain the details of read and write operations while in Select Strobe Mode.

2.5.5.1 Asynchronous Read Operations (Select Strobe Mode)

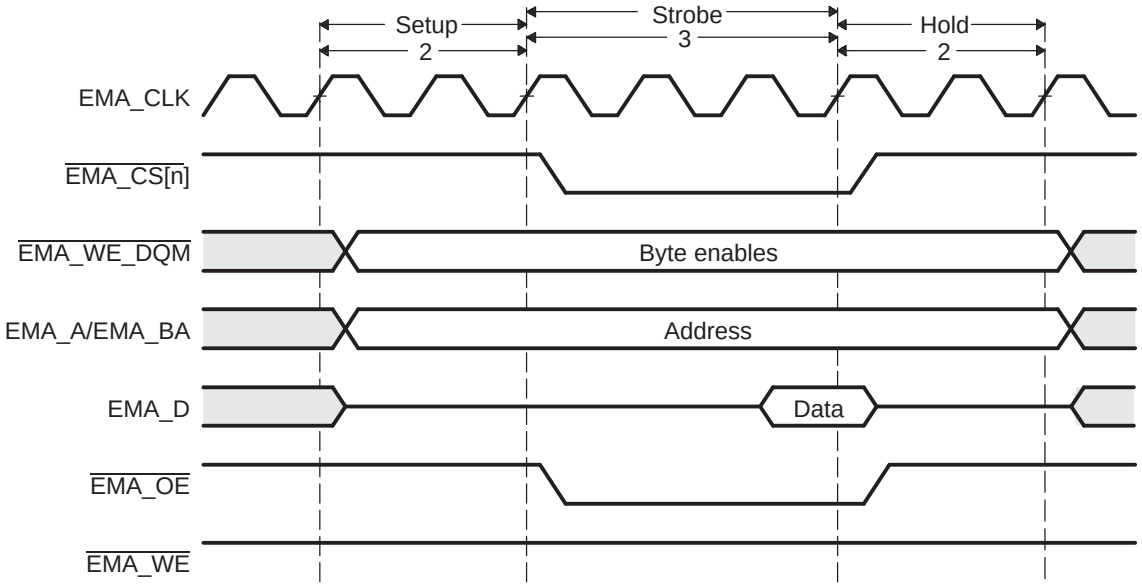
NOTE: During the entirety of an asynchronous read operation, the EMA_WE pin is driven high.

An asynchronous read is performed when any of the requesters mentioned in Section 2.2 request a read from the attached asynchronous memory. After the request is received, a read operation is initiated once it becomes the EMIFA's highest priority task, according to the priority scheme detailed in Section 2.12. In the event that the read request cannot be serviced by a single access cycle to the external device, multiple access cycles will be performed by the EMIFA until the entire request is fulfilled. The details of an asynchronous read operation in Select Strobe Mode are described in Table 21. Also, Figure 12 shows an example timing diagram of a basic read operation.

Table 21. Asynchronous Read Operation in Select Strobe Mode

Time Interval	Pin Activity in Select Strobe Mode
Turnaround period	Once the read operation becomes the highest priority task for the EMIFA, the EMIFA waits for the programmed number of turn-around cycles before proceeding to the setup period of the operation. The number of wait cycles is taken directly from the TA field of the asynchronous *n* configuration register (CE*n*CFG). There are two exceptions to this rule: • If the current read operation was directly proceeded by another read operation, no turn-around cycles are inserted. • If the current read operation was directly proceeded by a write operation and the TA field has been cleared to 0, one turn-around cycle will be inserted. After the EMIFA has waited for the turn-around cycles to complete, it again checks to make sure that the read operation is still its highest priority task. If so, the EMIFA proceeds to the setup period of the operation. If it is no longer the highest priority task, the EMIFA terminates the operation.
Start of the setup period	The following actions occur at the start of the setup period: • The setup, strobe, and hold values are set according to the R_SETUP, R_STROBE, and R_HOLD values in CE*n*CFG. • The address pins EMA_A and EMA_BA become valid and carry the values described in Section 2.5.1. • The EMA_WE_DQM pins become valid as byte enables.
Strobe period	The following actions occur during the strobe period of a read operation: 1. EMA_CS[n] (n = 2,3,4, or 5) and EMA_OE fall at the start of the strobe period 2. On the rising edge of the clock which is concurrent with the end of the strobe period: • EMA_CS[n] (n = 2,3,4, or 5) and EMA_OE rise • The data on the EMA_D bus is sampled by the EMIFA. In Figure 12, EMA_WAIT is inactive. If EMA_WAIT is instead activated, the strobe period can be extended by the external device to give it more time to provide the data. Section 2.5.7 contains more details on using the EMA_WAIT pin.
End of the hold period	At the end of the hold period: • The address pins EMA_A and EMA_BA become invalid • The EMA_EMA_WE_DQM pins become invalid The EMIFA may be required to issue additional read operations to a device with a small data bus width in order to complete an entire word access. In this case, the EMIFA immediately re-enters the setup period to begin another operation without incurring the turnaround cycle delay. The setup, strobe, and hold values are not updated in this case. If the entire word access has been completed, the EMIFA returns to its previous state unless another asynchronous request has been submitted and is currently the highest priority task. If this is the case, the EMIFA instead enters directly into the turnaround period for the pending read or write operation.

Figure 12. Timing Waveform of an Asynchronous Read Cycle in Select Strobe Mode



2.5.5.2 Asynchronous Write Operations (Select Strobe Mode)

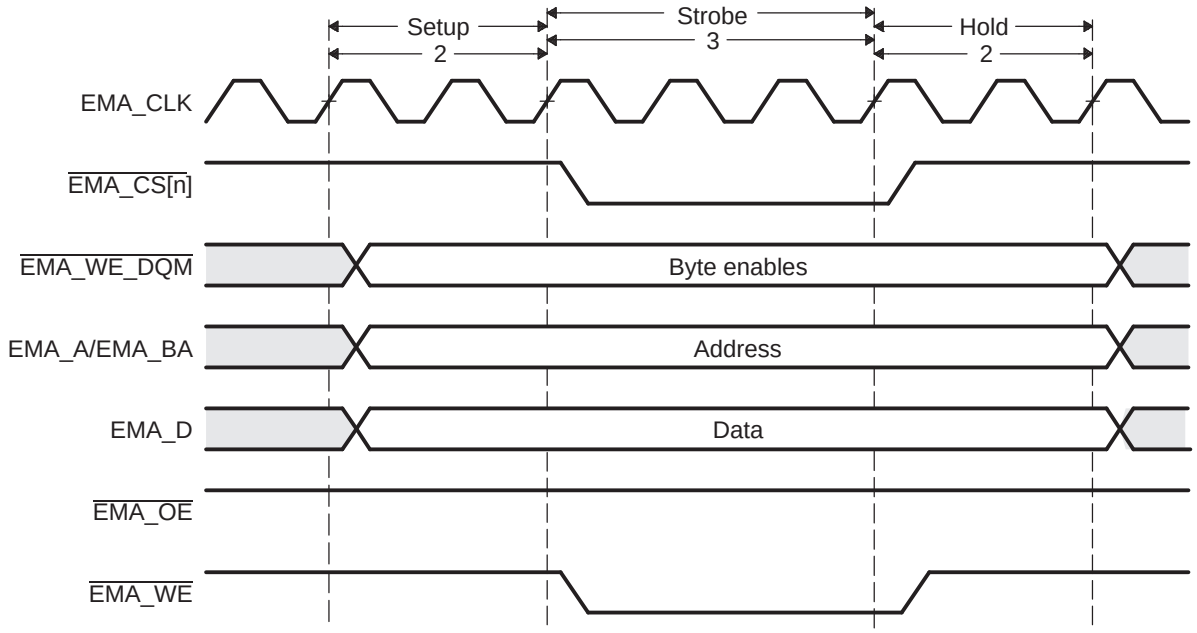
NOTE: During the entirety of an asynchronous write operation, the $\overline{\text{EMA\_OE}}$ pin is driven high.

An asynchronous write is performed when any of the requesters mentioned in Section 2.2 request a write to memory in the asynchronous bank of the EMIFA. After the request is received, a write operation is initiated once it becomes the EMIFA's highest priority task, according to the priority scheme detailed in Section 2.12. In the event that the write request cannot be serviced by a single access cycle to the external device, multiple access cycles will be performed by the EMIFA until the entire request is fulfilled. The details of an asynchronous write operation in Select Strobe Mode are described in Table 22. Also, Figure 13 shows an example timing diagram of a basic write operation.

Table 22. Asynchronous Write Operation in Select Strobe Mode

Time Interval	Pin Activity in Select Strobe Mode
Turnaround period	Once the write operation becomes the highest priority task for the EMIFA, the EMIFA waits for the programmed number of turnaround cycles before proceeding to the setup period of the operation. The number of wait cycles is taken directly from the TA field of the asynchronous *n* configuration register (CE*n*CFG). There are two exceptions to this rule: • If the current write operation was directly proceeded by another write operation, no turn-around cycles are inserted. • If the current write operation was directly proceeded by a read operation and the TA field has been cleared to 0, one turnaround cycle will be inserted. After the EMIFA has waited for the turnaround cycles to complete, it again checks to make sure that the write operation is still its highest priority task. If so, the EMIFA proceeds to the setup period of the operation. If it is no longer the highest priority task, the EMIFA terminates the operation.
Start of the setup period	The following actions occur at the start of the setup period: • The setup, strobe, and hold values are set according to the W_SETUP, W_STROBE, and W_HOLD values in CE*n*CFG. • The address pins EMA_A and EMA_BA and the data pins EMA_D become valid. The EMA_A and EMA_BA pins carry the values described in Section 2.5.1. • The $\overline{\text{EMA\_WE\_DQM}}$ pins become active as byte enables.
Strobe period	The following actions occur at the start of the strobe period of a write operation: • $\overline{\text{EMA\_CS[n]}}$ (n = 2,3,4, or 5) and $\overline{\text{EMA\_WE}}$ fall The following actions occur on the rising edge of the clock which is concurrent with the end of the strobe period: • $\overline{\text{EM\_CS[n]}}$ (n = 2,3,4, or 5) and $\overline{\text{EMA\_WE}}$ rise In Figure 13, EMA_WAIT is inactive. If EMA_WAIT is instead activated, the strobe period can be extended by the external device to give it more time to accept the data. Section 2.5.7 contains more details on using the EMA_WAIT pin.
End of the hold period	At the end of the hold period: • The address pins EMA_A and EMA_BA become invalid • The data pins become invalid • The $\overline{\text{EMA\_WE\_DQM}}$ pins become invalid The EMIFA may be required to issue additional write operations to a device with a small data bus width in order to complete an entire word access. In this case, the EMIFA immediately re-enters the setup period to begin another operation without incurring the turnaround cycle delay. The setup, strobe, and hold values are not updated in this case. If the entire word access has been completed, the EMIFA returns to its previous state unless another asynchronous request has been submitted and is currently the highest priority task. If this is the case, the EMIFA instead enters directly into the turn-around period for the pending read or write operation.

Figure 13. Timing Waveform of an Asynchronous Write Cycle in Select Strobe Mode



2.5.6 NAND Flash Mode

NAND Flash Mode is a submode of both Normal Mode and Select Strobe Mode. Chip select $\overline{\text{EM\_CS}}$[n] (n = 2, 3, 4, or 5) may be placed in NAND Flash mode by setting the CS*n*NAND (*n* = 2, 3, 4, or 5) bit in the NAND Flash control register (NANDFCR). Table 23 displays the bit fields present in NANDFCR and briefly describes their use.

When a chip select space is configured to operate in NAND Flash mode, the EMIFA hardware can calculate the error correction code (ECC) for each 512 byte data transfer to that chip select space. The EMIFA hardware will not generate the NAND access cycle, which includes the command, address, and data phases, necessary to complete a transfer to NAND Flash. All NAND Flash operations can be divided into single asynchronous cycles, and with the help of software the EMIFA can execute a complete NAND access cycle.

Table 23. Description of the NAND Flash Control Register (NANDFCR)

Parameter	Description
CS5ECC	**NAND Flash ECC state for $\overline{\text{EMA\_CS}}$[5].** • Set to 1 to start an ECC calculation for $\overline{\text{EMA\_CS}}$[5] • Cleared to 0 when NAND Flash 4 ECC register (NANDF4ECC) is read.
CS5NAND	**NAND Flash mode for $\overline{\text{EMA\_CS}}$[5].** • Set to 1 to enable NAND Flash mode for $\overline{\text{EMA\_CS}}$[5]
CS4ECC	**NAND Flash ECC state for $\overline{\text{EMA\_CS}}$[4].** • Set to 1 to start an ECC calculation for $\overline{\text{EMA\_CS}}$[4] • Cleared to 0 when NAND Flash 3 ECC register (NANDF3ECC) is read.
CS4NAND	**NAND Flash mode for $\overline{\text{EMA\_CS}}$[4].** • Set to 1 to enable NAND Flash mode for $\overline{\text{EMA\_CS}}$[4]
CS3ECC	**NAND Flash ECC state for $\overline{\text{EMA\_CS}}$[3].** • Set to 1 to start an ECC calculation for $\overline{\text{EMA\_CS}}$[3] • Cleared to 0 when NAND Flash 2ECC register (NANDF2ECC) is read.
CS3NAND	**NAND Flash mode for $\overline{\text{EMA\_CS}}$[3].** • Set to 1 to enable NAND Flash mode for $\overline{\text{EMA\_CS}}$[3]
CS2ECC	**NAND Flash ECC state for $\overline{\text{EMA\_CS}}$[2].** • Set to 1 to start an ECC calculation for $\overline{\text{EMA\_CS}}$[2] • Cleared to 0 when NAND Flash 1 ECC register (NANDF1ECC) is read.
CS2NAND	**NAND Flash mode for $\overline{\text{EMA\_CS}}$[2].** • Set to 1 to enable NAND Flash mode for $\overline{\text{EMA\_CS}}$[2]

2.5.6.1 Configuring for NAND Flash Mode

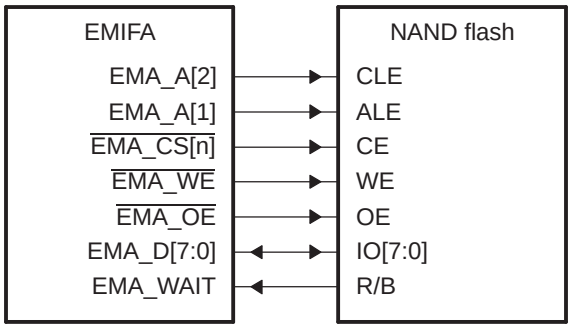
Similar to the asynchronous accesses previously described, the EMIFA's memory-mapped registers must be programmed appropriately to interface to a NAND Flash device. In addition to the fields listed in Table 15, the CS*n*NAND (*n* = 2,3,4, or 5) bit of the NAND Flash control register (NANDFCR) should be set to 1 to enter NAND Flash Mode. Note that the EW bit of CE*n*CFG should be cleared to avoid enabling the wait feature while in NAND Flash Mode.

2.5.6.2 Connecting to NAND Flash

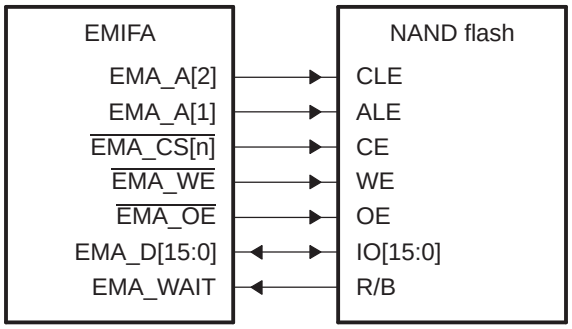
Figure 14 shows the EMIFA external pins used to interface with a NAND Flash device. EMIFA address lines are used to drive the NAND Flash device's command latch enable (CLE) and address latch enable (ALE) signals. Any EMIFA address lines may be used to drive the CLE and ALE signals of the NAND Flash.

NOTE: The EMIFA will not control the NAND Flash device's write protect pin. The write protect pin must be controlled outside of the EMIFA.

Figure 14. EMIFA to NAND Flash Interface



a) Connection to 8−bit NAND device



b) Connection to 16−bit NAND device

2.5.6.3 Driving CLE and ALE

As stated in Section 2.5.1, the EMIFA always drives the least significant bit of a 32-bit word address on EMA_A[0]. This functionality must be considered when attempting to drive the offset lines connected to CLE and ALE to the appropriate state.

For example, if using EMA_A[2] and EMA_A[1] to connect to CLE and ALE, respectively, the following offsets should be added to EMIFA base address:

- 0000 0000h to drive CLE and ALE low
- 0000 0010h to drive CLE high and ALE low
- 0000 0008h to drive CLE low and ALE high

2.5.6.4 NAND Read and Program Operations

A NAND Flash access cycle is composed of a command, address, and data phase. The EMIFA will not automatically generate these three phases to complete a NAND access with one transfer request. To complete a NAND access cycle, multiple single asynchronous access cycles must be completed by the EMIFA. Software must be used to request the appropriate asynchronous accesses to complete a NAND Flash access cycle. This software must be developed to the specification of the chosen NAND Flash device.

Since NAND operations are divided into single asynchronous access cycles, the chip select signal will not remain activated for the duration of the NAND operation. Instead, the chip select signal will deactivate between each asynchronous access cycle. For this reason, the EMIFA does not support NAND Flash devices that require the chip select signal to remain low during the t_R time for a read. See Section 2.5.6.8 for workaround.

Care must be taken when performing a NAND read or write operation via the EDMA controller. See Section 2.5.6.5 for more details.

NOTE: The EMIFA does not support NAND Flash devices that require the chip select signal to remain low during the t_R time for a read. See Section 2.5.6.8 for workaround.

2.5.6.5 NAND Data Read and Write via EDMA Controller

When performing NAND accesses, the EDMA controller is most efficiently used for the data phase of the access. The command and address phases of the NAND access require only a few words of data to be transferred and therefore do not take advantage of the EDMA controller's ability to transfer larger quantities of data with a single request. In this section we will focus on using the EDMA controller for the data phase of a NAND access.

There are two conditions that require care to be taken when performing NAND reads and writes via the EDMA controller. These are:

- The address lines used to drive CLE and ALE signals must be driven low
- The EMIFA does not support constant addressing mode

Since the EMIFA does not support a constant addressing mode, when programming the EDMA, a linear incrementing address mode must be used. When using a linear incrementing address mode, if the CLE and ALE are driven by EM_A[2] and EM_A[1], respectively, care must be taken not to increase the address into a range that drives CLE and/or ALE high. To prevent the address from incrementing into a range that drives CLE and/or ALE high, the EDMA ACNT, BCNT, SIDX, DIDX, and synchronization type must be programmed appropriately. Following is an example configuration of EDMA controller when EM_A[2] is connected to CLE and EM_A[1] is connected to ALE.

EDMA setup for a NAND Flash data read:

- ACNT ≤ 8 bytes (this can also be set to less than or equal to the external data bus width)
- BCNT = transfer size in bytes/ACNT
- SIDX (source index) = 0
- DIDX (destination index) = ACNT
- AB synchronized

EDMA setup for a NAND Flash data write:

- ACNT ≤ 8 bytes (this can also be set to less than or equal to the external data bus width)
- BCNT = transfer size in bytes/ACNT
- SIDX (source index) = ACNT
- DIDX (destination index) = 0
- AB synchronized

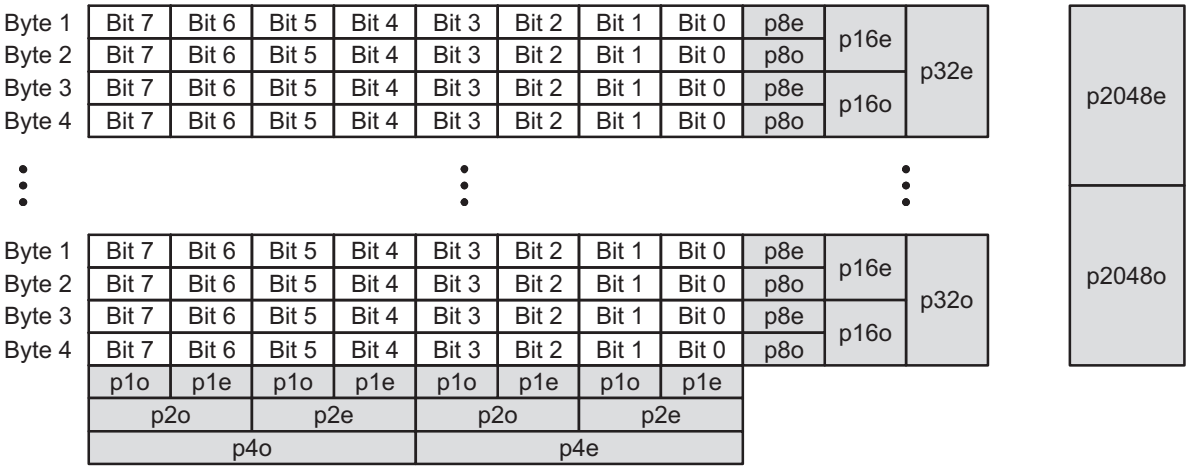
2.5.6.6 ECC Generation

2.5.6.6.1 1-Bit ECC

If the CS*n*NAND (*n* = 2, 3, 4, or 5) bit in the NAND Flash control register (NANDFCR) is set to 1, the EMIFA supports ECC calculation for up to 512 bytes for the corresponding chip select. To perform the ECC calculation, the CS*n*ECC (*n* = 2, 3, 4, or 5) bit in NANDFCR must be set to 1. It is the responsibility of the software to start the ECC calculation by writing to the CS*n*ECC (*n* = 2, 3, 4, or 5) bit prior to issuing a write or read to NAND Flash. It is also the responsibility of the software to read the calculated ECC from the NAND Flash *m* ECC register (NANDF*m*ECC) (*m* = 1, 2, 3, or 4) once the transfer to NAND Flash has completed. If the software writes or reads more than 512 bytes, the ECC will be incorrect. Reading the NAND*m*ECC (*m* = 1, 2, 3, or 4) clears the CS*n*ECC (*n* = 2, 3, 4, or 5) bit in NANDFCR. The NANDF*m*ECC (*m* = 1, 2, 3, or 4) is cleared upon writing a 1 to the CS*n*ECC (*n* = 2, 3, 4, or 5) bit. Figure 15 shows the algorithm used to calculate the ECC value for an 8-bit NAND Flash.

For an 8-bit NAND Flash p1o through p4e are column parities and p8e through p2048o are row parities. Similarly, the algorithm can be extended to a 16-bit NAND Flash. For a 16-bit NAND Flash p1o through p8e are column parities and p16e through p2048o are row parities. The software must ignore the unwanted parity bits if ECC is desired for less than 512 bytes of data. For example. p2048e and p2048o are not required for ECC on 256 bytes of data. Similarly, p1024e, p1024o, p2048e, and p2048o are not required for ECC on 128 bytes of data.

Figure 15. ECC Value for 8-Bit NAND Flash



2.5.6.6.2 4-Bit ECC

The EMIFA supports 4-bit ECC only for 8-bit NAND Flash. In NAND mode, if the NAND Flash 4-bit ECC start bit (4BITECC_START) in the in the NAND Flash control register (NANDFCR) is set, the EMIFA calculates 4-bit ECC for the selected chip select. Only one chip select can be selected for the 4-bit ECC calculation at one time. The selection of the chip select is done by programming the 4-bit ECC CS select bit field (4BITECCSEL) in the NAND Flash control register (NANDFCR). The calculated parity (for writes) and syndrome (for reads) can be read from the NAND Flash 4-Bit ECC 1-4 registers (NAND4BITECC[4:1]). The 4-bit ECC start bit (4BITECC_START) is cleared upon reading any of the NAND Flash 4-bit ECC 1-4 registers (NAND4BITECC[4:1]). The NAND Flash 4-Bit ECC 1-4 registers are cleared upon writing one to the 4-bit ECC start bit (4BITECC_START).

The 4-bit ECC algorithm works on a 10-bit data bus. Since the 4-bit ECC is only used for an 8-bit NAND Flash, the EMIFA zeros the upper two bits. However, the parity and the syndrome value read from the NAND Flash 4-bit ECC 1-4 registers (NAND4BITECC[4:1]) are 10 bits wide. It is the responsibility of software to convert 10-bit parity values to 8 bits before writing to the spare location of the NAND Flash after a write operation. Similarly, it is the responsibility of the software to convert the 8-bit parity values read from the spare location of the NAND Flash after a read operation, to 10 bits before writing the NAND Flash 4-bit ECC load register (NAND4BITECCLOAD).

At the end of the syndrome calculation after read, the error address and the error value can be calculated by setting the address and error value calculation start bit (4BITECC_ADD_CALC_START) in the NAND Flash control register (NANDFCR). The end of address calculation is flagged by the 4-bit ECC correction state field (ECC_STATE) in the NAND Flash status register (NANDFSR). The number of errors can be read from the 4-bit number of errors field (ECC_ERRNUM) in the NAND Flash status register (NANDFSR). The error address value can be read from the NAND Flash error address 1-2 registers (NANDERRADD[2:1]). The error value can be read from the NAND Flash error value 1-2 registers (NANDERRVAL[2:1]). The address and error value start bit (4BITECC_ADD_CALC_START) is cleared upon reading any of the NAND Flash error address 1-2 registers (NANDERRADD[2:1]) or the NAND Flash error value 1-2 registers (NANDERRVAL[2:1]). The EMIFA registers the syndrome value internally before the error address and error value calculation. Therefore, a new read operation can be performed simultaneously with the error address calculation.

The EMIF supports 4-bit ECC calculation up to 518 bytes. The software needs to follow the following procedure for 4-bit ECC calculation:

For writes:

1. Set the 4BITECC_START bit in the NAND Flash control register (NANDFCR) to 1.
2. Write 518 bytes of data to the NAND Flash.
3. Read the parity from the NAND Flash 4-Bit ECC 1-4 registers (NAND4BITECC[4:1]).
4. Convert the 10-bit parity values to 8-bits. All 10-bit parity values can be concatenated together with ECC value 1 (4BITECCVAL1) as LSB and ECC value 8 (4BITECCVAL8) as MSB. Then the concatenated value can be broken down into ten 8-bit values.
5. Store the parity to spare location in the NAND Flash.

For reads:

1. Set the 4BITECC_START bit in the NAND Flash control register (NANDFCR) to 1.
2. Read 518 bytes of data from the NAND Flash.
3. Clear the 4BITECC_START bit in NANDFCR by reading any of the NAND Flash 4-bit ECC registers.
4. Read the parity stored in the spare location in the NAND Flash.
5. Convert the 8-bit parity values to 10-bits. Reverse of the conversion that was done during writes.
6. Write the parity values in the NAND Flash 4-bit ECC load register (NAND4BITECCLOAD). Write each parity value one at a time starting from 4BITECCVAL8 down to 4BITECCVAL1.
7. Perform a dummy read to the NAND Flash status register (NANDFSR). This is only required to ensure time for syndrome calculation after writing the ECC values in step 6.
8. Read the syndrome from the NAND Flash 4-bit ECC 1-4 registers (NAND4BITECC[4:1]). A syndrome value of 0 means no bit errors. If the syndrome is non-zero, continue with step 9.
9. Set the 4BITECC_ADD_CALC_START bit in the NAND Flash control register (NANDFCR) to 1.
10. Start another read from NAND, if required (a new thread from step 1).
11. Wait for the 4-bit ECC correction state field (ECC_STATE) in the NAND Flash status register (NANDFSR) to be equal to 1, 2h, or 3h.
12. The number of errors can be read from the 4-bit number of errors field (ECC_ERRNUM) in the NAND Flash status register (NANDFSR).
13. Read the error address from the NAND Flash error address 1-2 registers (NANDERRADD[2:1]). Address for the error word is equal to (total_words_read + 7 - address_value). For 518 bytes, the address will be equal to (525 - address_value).
14. Read the error value from the NAND Flash error value 1-2 registers (NANDERRVAL[2:1]). Errors can be corrected by XORing the error word with the error value from the NAND Flash error value 1-2 registers (NANDERRVAL[2:1]).

2.5.6.7 NAND Flash Status Register (NANDFSR)

The NAND Flash status register (NANDFSR) indicates the raw status of the EMA_WAIT pin while in NAND Flash Mode. The EMA_WAIT pin should be connected to the NAND Flash device's R/$\overline{B}$ signal, so that it indicates whether or not the NAND Flash device is busy. During a read, the R/$\overline{B}$ signal will transition and remain low while the NAND Flash retrieves the data requested. Once the R/$\overline{B}$ signal transitions high, the requested data is ready and should be read by the EMIFA. During a write/program operation, the R/$\overline{B}$ signal transitions and remains low while the NAND Flash is programming the Flash with the data it has received from the EMIFA. Once the R/$\overline{B}$ signal transitions high, the data has been written to the Flash and the next phase of the transaction may be performed. From this explanation, you can see that the NAND Flash status register is useful to the software for indicating the status of the NAND Flash device and determining when to proceed to the next phase of a NAND Flash operation.

When a rising edge occurs on the EMA_WAIT pin, the EMIFA sets the WR (Wait Rise) bit in the EMIFA interrupt raw register (INTRAW). Therefore, the EMIFA Wait Rise interrupt may be used to indicate the status of the NAND Flash device. The WP*n* bit in the asynchronous wait cycle configuration register (AWCC) does not affect the NAND Flash status register (NANDFSR) or the WR bit in INTRAW. See Section 2.8 for more a detailed description of the wait rise interrupt.

2.5.6.8 Interfacing to a Non-CE Don't Care NAND Flash

As explained in Section 2.5.6.4, the EMIFA does not support NAND Flash devices that require the chip select signal to remain low during the t_R time for a read. One way to work around this limitation is to use a GPIO pin to drive the $\overline{CE}$ signal of the NAND Flash device. If this work around is implemented, software will configure the selected GPIO to be low, then begin the NAND Flash operation, starting with the command phase. Once the NAND Flash operation has completed the software can then configure the selected GPIO to be high.

2.5.7 Extended Wait Mode and the EMA_WAIT Pin

The EMIFA supports the Extend Wait Mode. This is a mode in which the external asynchronous device may assert control over the length of the strobe period. The Extended Wait Mode can be entered by setting the EW bit in the asynchronous *n* configuration register (CE*n*CFG) (*n* = 2,3,4, or 5). When this bit is set, the EMIFA monitors the EMA_WAIT pin to determine if the attached device wishes to extend the strobe period of the current access cycle beyond the programmed number of clock cycles.

When the EMIFA detects that the EMA_WAIT pin has been asserted, it will begin inserting extra strobe cycles into the operation until the EMA_WAIT pin is deactivated by the external device. The EMIFA will then return to the last cycle of the programmed strobe period and the operation will proceed as usual from this point. Please refer to the device data manual for details on the timing requirements of the EMA_WAIT signal.

The EMA_WAIT pin cannot be used to extend the strobe period indefinitely. The programmable MAX_EXT_WAIT field in the asynchronous wait cycle configuration register (AWCC) determines the maximum number of EMA_CLK cycles the strobe period may be extended beyond the programmed length. When the counter expires, the EMIFA proceeds to the hold period of the operation regardless of the state of the EMA_WAIT pin. The EMIFA can also generate an interrupt upon expiration of this counter. See Section 2.8.1 for details on enabling this interrupt.

For the EMIFA to function properly in the Extended Wait mode, the WP*n* bit of AWCC must be programmed to match the polarity of the EMA_WAIT pin. In its reset state of 1, the EMIFA will insert wait cycles when the EMA_WAIT pin is sampled high. When set to 0, the EMIFA will insert wait cycles only when EMA_WAIT is sampled low. This programmability allows for a glueless connection to larger variety of asynchronous devices.

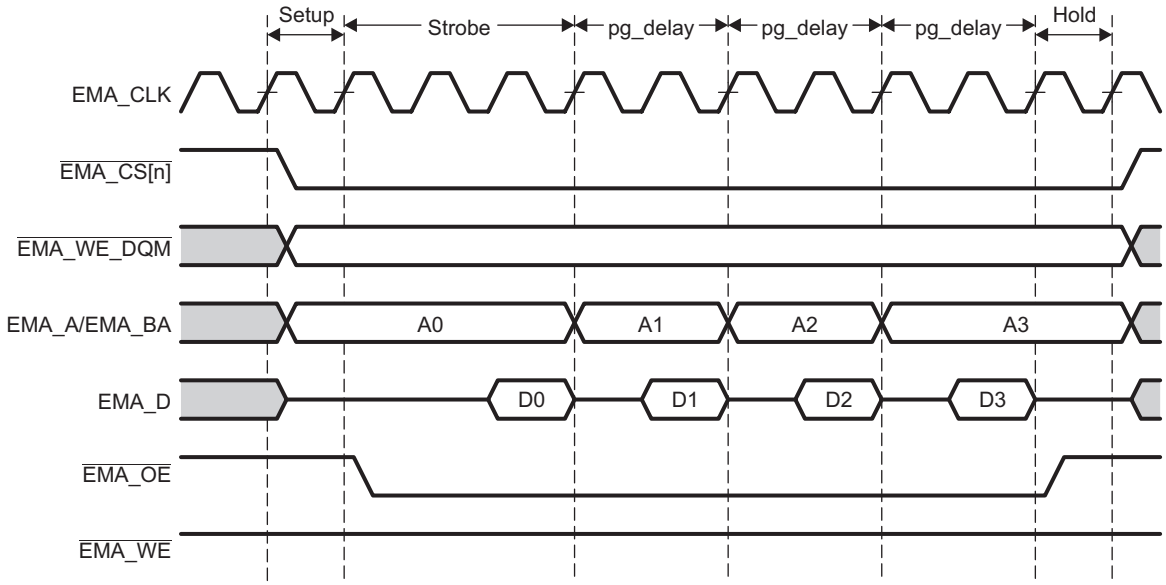
Finally, a restriction is placed on the strobe period timing parameters when operating in Extended Wait mode. Specifically, the W_STROBE and R_STROBE fields must not be set to 0 for proper operation.

2.5.8 NOR Flash Page Mode

EMIFA supports Page mode reads for NOR Flash on its asynchronous memory chip selects. This mode can be enabled by writing a 1 to the CS*n*_PG_MD_EN (*n*=2,3,4, or 5) field in the Page Mode Control register for the chip select in consideration. Whenever Page Mode for reads is enabled for a particular chip select, the page size for the device connected must also be programmed in the CS*n*_PG_SIZE field of the Page Mode Control register. The address change to valid read data available timing must be programmed in the CS*n*_PG_DEL field of the Page Control register. All other asynchronous memory timings must be programmed in the asynchronous configuration register (CE*n*CFG). See Figure 16 for read in asynchronous page mode.

NOTE: The Extended Wait mode and the Select Strobe mode must be disabled when using the asynchronous interface in Page mode.

Figure 16. Asynchronous Read in Page Mode



2.6 Data Bus Parking

The EMIFA always drives the data bus to the previous write data value when it is idle. This feature is called data bus parking. Only when the EMIFA issues a read command to the external memory does it stop driving the data bus. After the EMIFA latches the last read data, it immediately parks the data bus again.

The one exception to this behavior occurs after performing an asynchronous read operation while the EMIFA is in the self-refresh state. In this situation, the read operation is not followed by the EMIFA parking the data bus. Instead, the EMIFA tri-states the data bus. Therefore, it is not recommended to perform asynchronous read operations while the EMIFA is in the self-refresh state, in order to prevent floating inputs on the data bus. External pull-ups, such as 10kΩ resistors, should be placed on the 16 EMIFA data bus pins (which do not have internal pull-ups) if it is required to perform reads in this situation. The precise resistor value should be chosen so that the worst case combined off-state leakage currents do not cause the voltage levels on the associated pins to drop below the high-level input voltage requirement.

More information about the self-refresh state can be found in Section 2.4.7.

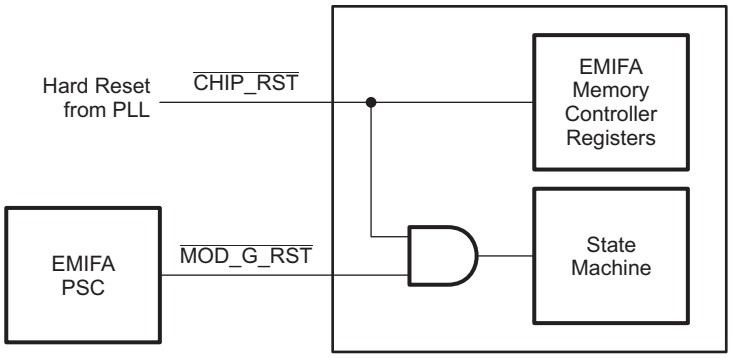
2.7 Reset and Initialization Considerations

The EMIFA 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 EMIFA memory controller's memory-mapped registers. The MOD_G_RST resets the state machine only. If the EMIFA 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 EMIFA memory controller is in the reset state, other masters may hang. Following the rising edge of CHIP_RST or MOD_G_RST, the EMIFA memory controller immediately begins its initialization sequence. Command and data stored in the EMIFA memory controller FIFOs are lost. Table 24 describes the different methods for asserting each reset signal. Figure 17 shows the EMIFA memory controller reset diagram.

Table 24. Reset Sources

Reset Signal	Reset Source
CHIP_RST	Hardware/ Device Reset
MOD_G_RST	Power and Sleep Controller

Figure 17. EMIFA Reset Block Diagram



The EMIFA and its registers will be reset when any of the following events occur:

1. The RESET pin on the device is asserted
2. An emulator reset is initiated through Code Composer Studio

In the first case, the EMIFA will exit the reset state when RESET is released and after the PLL controller releases the entire device from reset. In the second case, the EMIFA will exit the reset state immediately after the emulator reset is complete.

In both cases, the EMIFA automatically begins running the SDRAM initialization sequence described in Section 2.4.4 after coming out of reset. Even though the initialization procedure is automatic, a special procedure, found in Section 2.4.5 must still be followed.

2.8 Interrupt Support

The EMIFA supports a single interrupt to the CPU. Section 2.8.1 details the generation and internal masking of EMIFA interrupts, and Section 2.8.2 describes how the EMIFA interrupts are sent to the CPU.

2.8.1 Interrupt Events

There are three conditions that may cause the EMIFA to generate an interrupt to the CPU. These conditions are:

- A rising edge on the EMA_WAIT signal (wait rise interrupt)
- An asynchronous time out
- Usage of unsupported addressing mode (line trap interrupt)

The wait rise interrupt occurs when a rising edge is detected on EMA_WAIT signal. This interrupt generation is not affected by the WP*n* bit in the asynchronous wait cycle configuration register (AWCC). The asynchronous time out interrupt condition occurs when the attached asynchronous device fails to deassert the EMA_WAIT pin within the number of cycles defined by the MAX_EXT_WAIT bit in AWCC (this happens only in extended wait mode). EMIFA supports only linear incrementing and cache line wrap addressing modes . If an access request for an unsupported addressing mode is received, the EMIFA will set the LT bit in the EMIFA interrupt raw register (INTRAW) and treat the request as a linear incrementing request.

Only when the interrupt is enabled by setting the appropriate bit (WR_MASK_SET/AT_MASK_SET/LT_MASK_SET) in the EMIFA interrupt mask set register (INTMSKSET) to 1, will the interrupt be sent to the CPU. Once enabled, the interrupt may be disabled by writing a 1 to the corresponding bit in the EMIFA interrupt mask clear register (INTMSKCLR). The bit fields in both the INTMSKSET and INTMSKCLR may be used to indicate whether the interrupt is enabled. When the interrupt is enabled, the corresponding bit field in both the INTMSKSET and INTMSKCLR will have a value of 1; when the interrupt is disabled, the corresponding bit field will have a value of 0.

The EMIFA interrupt raw register (INTRAW) and the EMIFA interrupt mask register (INTMSK) indicate the status of each interrupt. The appropriate bit (WR/AT/LT) in INTRAW is set when the interrupt condition occurs, whether or not the interrupt has been enabled. However, the appropriate bit (WR_MASKED/AT_MASKED/LT_MASKED) in INTMSK is set only when the interrupt condition occurs and the interrupt is enabled. Writing a 1 to the bit in INTRAW clears the INTRAW bit as well as the corresponding bit in INTMSK. Table 25 contains a brief summary of the interrupt status and control bit fields. See Section 3 for complete details on the register fields.

Table 25. Interrupt Monitor and Control Bit Fields

Register Name	Bit Name	Description
EMIF interrupt raw register (INTRAW)	WR	This bit is set when an rising edge on the EMA_WAIT signal occurs. Writing a 1 clears the WR bit as well as the WR_MASKED bit in INTMSK.
	AT	This bit is set when an asynchronous timeout occurs. Writing a 1 clears the AT bit as well as the AT_MASKED bit in INTMSK.
	LT	This bit is set when an unsupported addressing mode is used. Writing a 1 clears LT bit as well as the LT_MASKED bit in INTMSK.
EMIF interrupt mask register (INTMSK)	WR_MASKED	This bit is set only when a rising edge on the EMA_WAIT signal occurs and the interrupt has been enabled by writing a 1 to the WR_MASK_SET bit in INTMSKSET.
	AT_MASKED	This bit is set only when an asynchronous timeout occurs and the interrupt has been enabled by writing a 1 to the AT_MASK_SET bit in INTMSKSET.
	LT_MASKED	This bit is set only when line trap interrupt occurs and the interrupt has been enabled by writing a 1 to the LT_MASK_SET bit in INTMSKSET.
EMIF interrupt mask set register (INTMSKSET)	WR_MASK_SET	Writing a 1 to this bit enables the wait rise interrupt.
	AT_MASK_SET	Writing a 1 to this bit enables the asynchronous timeout interrupt.
	LT_MASK_SET	Writing a 1 to this bit enables the line trap interrupt.
EMIF interrupt mask clear register (INTMSKCLR)	WR_MASK_CLR	Writing a 1 to this bit disables the wait rise interrupt.
	AT_MASK_CLR	Writing a 1 to this bit disables the asynchronous timeout interrupt.
	LT_MASK_CLR	Writing a 1 to this bit disables the line trap interrupt.

2.8.2 Interrupt Multiplexing

For details on EMIFA interrupt multiplexing, see your device-specific *System Reference Guide*.

2.8.3 Interrupt Processing

For details on EMIFA interrupt processing, see your device-specific *System Reference Guide*.

For more details on the CPU's NMI interrupt, see the *TMS320C674x CPU and Instruction Set Reference Guide* (SPRUFE8).

2.9 EDMA Event Support

EMIFA 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 Pin Multiplexing

For details on EMIFA pin multiplexing, see your device-specific *System Reference Guide* and your device-specific data manual.

2.11 Memory Map

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

2.12 Priority and Arbitration

Section 2.2 of this document describes the external prioritization and arbitration among requests from different sources within the SoC. The result of this external arbitration is that only one request is presented to the EMIFA at a time. Once the EMIFA completes a request, the external arbiter then provides the EMIFA with the next pending request.

Internally, the EMIFA undertakes memory device transactions according to a strict priority scheme. The highest priority events are:

- A device reset.
- A write to any of the three least significant bytes of the SDRAM configuration register (SDCR).

Either of these events will cause the EMIFA to immediately commence its initialization sequence as described in Section 2.4.4.

Once the EMIFA has completed its initialization sequence, it performs memory transactions according to the following priority scheme (highest priority listed first):

1. If the EMIFA's backlog refresh counter is at the Refresh Must urgency level, the EMIFA performs multiple SDRAM auto refresh cycles until the Refresh Release urgency level is reached.
2. If an SDRAM or asynchronous read has been requested, the EMIFA performs a read operation.
3. If the EMIFA's backlog refresh counter is at the Refresh Need urgency level, the EMIFA performs an SDRAM auto refresh cycle.
4. If an SDRAM or asynchronous write has been requested, the EMIFA performs a write operation.
5. If the EMIFA's backlog refresh counter is at the Refresh May or Refresh Release urgency level, the EMIFA performs an SDRAM auto refresh cycle.
6. If the value of the SR bit in SDCR has been set to 1, the EMIFA will enter the self-refresh state as described in Section 2.4.7.

After taking one of the actions listed above, the EMIFA then returns to the top of the priority list to determine its next action.

Because the EMIFA does not issue auto-refresh cycles when in the self-refresh state, the above priority scheme does not apply when in this state. See Section 2.4.7 for details on the operation of the EMIFA when in the self-refresh state.

2.13 System Considerations

This section describes various system considerations to keep in mind when operating the EMIFA.

2.13.1 Asynchronous Request Times

In a system that interfaces to both SDRAM and asynchronous memory, the asynchronous requests must not take longer than the smaller of the following two values:

- t_{RAS} (typically 120 μs) - to avoid violating the maximum time allowed between issuing an ACTV and PRE command to the SDRAM.
- $t_{Refresh\ Rate}$ × 11 (typically 15.7 μs × 11 = 172.7 μs) - to avoid refresh violations on the SDRAM.

  The length of an asynchronous request is controlled by multiple factors, the primary factor being the number of access cycles required to complete the request. For example, an asynchronous request for 4 bytes will require four access cycles using an 8-bit data bus and only two access cycle using a 16-bit data bus. The maximum request size that the EMIFA can be sent is 16 words, therefore the maximum number of access cycles per memory request is 64 when the EMIFA is configured with an 8-bit data bus. The length of the individual access cycles that make up the asynchronous request is determined by the programmed setup, strobe, hold, and turnaround values, but can also be extended with the assertion of the EMA_WAIT input signal up to a programmed maximum limit. It is up to the user to make sure that an entire asynchronous request does not exceed the timing values listed above when also interfacing to an SDRAM device. This can be done by limiting the asynchronous timing parameters.

2.13.2 Cache Fill Requests

The CPU can run code from either internal or external memory. When running code from external memory, the CPU's program cache is periodically filled with eight words (32-bytes) through a dedicated port to the EMIFA. Two system level concerns arise when filling the program cache from the EMIFA.

First, the program cache fills have the possibility of being locked out from accessing the EMIFA by a stream of higher priority requests. Therefore, care should be taken when issuing persistent requests to the EMIFA from a source such which is a high priority requester.

Second, requests to the EMIFA from the other sources risk missing their deadlines while a program cache fill from the EMIFA is in progress. This is because all other EMIFA accesses are held pending while the program cache is filled. The worst-case scenario that can arise is when a requester submits a request immediately after a program cache fill request has begun. The system should be analyzed to make sure that this worst-case request delay is acceptable.

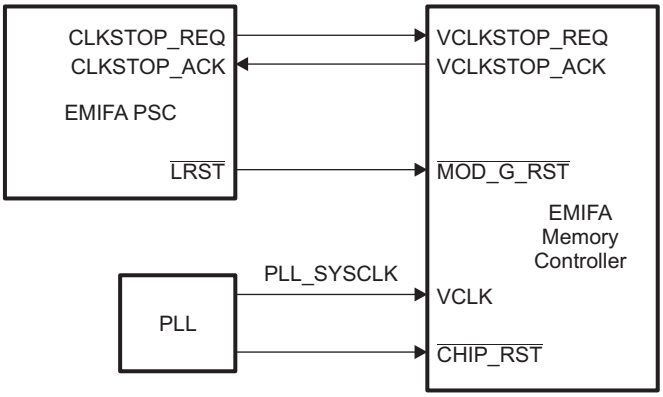
2.14 Power Management

Power dissipation from the EMIFA memory controller may be managed by following methods:

- Self-refresh mode
- Power-down mode
- Gating input clocks to the module off

Gating input clocks off to the EMIFA 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 EMIFA memory controller through the use of the Power and Sleep Controller (PSC) and the PLL controller. Figure 18 shows the connections between the EMIFA memory controller, PSC, and PLL. Before gating clocks off, the EMIFA memory controller must place the SDR SDRAM memory in self-refresh mode. If the external memory requires a continuous clock, the clock provided by the 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 EMIFA memory controller clocks.

Figure 18. EMIFA PSC Block Diagram



2.14.1 Power Management Using Self Refresh Mode

The EMIFA 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 2.4.7 for more details on placing the EMIFA into the self-refresh state.

2.14.2 Power Management Using Power Down Mode

In case of power down, to lower the power consumption, EMIFA drives EMA_SDCKE low. EMA_SDCKE goes high when there is a need to send refresh (REFR) commands, after which EMA_SDCKE is again driven low. EMA_SDCKE remains low until any request arrives. Refer to Section 2.4.8 for more details on placing EMIFA in power down mode.

2.14.3 Power Management Using Clock Stop

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

- Enable
- Auto Sleep
- Auto Wake
- Sync Reset

After the EMIFA clock is enabled, by default it is in the enable state. EMIFA can be put to auto sleep state, when the clock is to be gated off. Auto Wake brings back EMIFA to the enable state from the auto sleep state.

2.14.3.1 Auto Sleep and Auto Wake

To achieve maximum power savings EMIFA core clock should be gated off. EMIFA memory controller can make use of auto sleep and auto wake to achieve clock gating. Following describes the procedure to be followed to put EMIFA memory controller in auto sleep state:

- EMIFA should be put to self refresh mode before stopping the clock. Refer Section 2.4.7 for details on self refresh mode. The EMIFA memory controller will complete any outstanding accesses and backlogged refresh cycles and then place the EMIFA memory in self-refresh mode.
- Then, program the LPSC of EMIFA for auto sleep, to gate off the clocks.

Register and memory access requests are honored while EMIFA is in auto sleep state. When EMIFA 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, EMIFA switches between auto sleep and enable states. To bring EMIFA back to the enable state, auto wake can be used. Following procedure is followed for performing auto wake.

- Program the LPSC of EMIFA for auto wake.
- Bring EMIFA out of self-refresh. Refer Section 2.4.7 for details on self refresh mode.

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

2.14.3.2 Sync Reset and Enable

Sync reset of EMIFA through the LPSC does not reset the EMIFA registers or memory. Thus EMIFA LPSC sync reset behavior is similar to EMIFA LPSC auto sleep, except that register or memory requests are not honored by EMIFA. Following is the procedure to put EMIFA in sync reset state:

- EMIFA should be put to self refresh mode before stopping the clock. Refer Section 2.4.7 for details on self refresh mode. The EMIFA memory controller will complete any outstanding accesses and backlogged refresh cycles and then place the EMIFA memory in self-refresh mode.
- Then, program the LPSC of EMIFA to Sync-Reset state.

On sync reset, requests to EMIFA are not honored. To bring EMIFA back to the enable state, use the following enable procedure:

- Program the LPSC of EMIFA to enter enable state.
- Bring EMIFA out of self-refresh. Refer Section 2.4.7 for details on self refresh mode.

Now EMIFA memory controller is in the enable state and continues with normal operation.

2.15 Emulation Considerations

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

3 Registers

The external memory interface (EMIFA) is controlled by programming its internal memory-mapped registers (MMRs). Table 26 lists the memory-mapped registers for the EMIFA.

NOTE: All EMIFA MMRs, except SDCR, support only word (32-bit) accesses. Performing a byte (8-bit) or halfword (16-bit) write to these registers results in undefined behavior. The SDCR is byte writable to allow the setting of the SR, PD and PDWR bits without triggering the SDRAM initialization sequence.

The EMIFA 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 EMIFA, see your device-specific data manual.

Table 26. External Memory Interface (EMIFA) Registers

Offset	Acronym	Register Description	Section
0h	MIDR	Module ID Register	Section 3.1
4h	AWCC	Asynchronous Wait Cycle Configuration Register	Section 3.2
8h	SDCR	SDRAM Configuration Register	Section 3.3
Ch	SDRCR	SDRAM Refresh Control Register	Section 3.4
10h	CE2CFG	Asynchronous 1 Configuration Register	Section 3.5
14h	CE3CFG	Asynchronous 2 Configuration Register	Section 3.5
18h	CE4CFG	Asynchronous 3 Configuration Register	Section 3.5
1Ch	CE5CFG	Asynchronous 4 Configuration Register	Section 3.5
20h	SDTIMR	SDRAM Timing Register	Section 3.6
3Ch	SDSRETR	SDRAM Self Refresh Exit Timing Register	Section 3.7
40h	INTRAW	EMIFA Interrupt Raw Register	Section 3.8
44h	INTMSK	EMIFA Interrupt Mask Register	Section 3.9
48h	INTMSKSET	EMIFA Interrupt Mask Set Register	Section 3.10
4Ch	INTMSKCLR	EMIFA Interrupt Mask Clear Register	Section 3.11
60h	NANDFCR	NAND Flash Control Register	Section 3.12
64h	NANDFSR	NAND Flash Status Register	Section 3.13
68h	PMCR	Page Mode Control Register	Section 3.14
70h	NANDF1ECC	NAND Flash 1 ECC Register (CS2 Space)	Section 3.15
74h	NANDF2ECC	NAND Flash 2 ECC Register (CS3 Space)	Section 3.15
78h	NANDF3ECC	NAND Flash 3 ECC Register (CS4 Space)	Section 3.15
7Ch	NANDF4ECC	NAND Flash 4 ECC Register (CS5 Space)	Section 3.15
BCh	NAND4BITECCLOAD	NAND Flash 4-Bit ECC Load Register	Section 3.16
C0h	NAND4BITECC1	NAND Flash 4-Bit ECC Register 1	Section 3.17
C4h	NAND4BITECC2	NAND Flash 4-Bit ECC Register 2	Section 3.18
C8h	NAND4BITECC3	NAND Flash 4-Bit ECC Register 3	Section 3.19
CCh	NAND4BITECC4	NAND Flash 4-Bit ECC Register 4	Section 3.20
D0h	NANDERRADD1	NAND Flash 4-Bit ECC Error Address Register 1	Section 3.21
D4h	NANDERRADD2	NAND Flash 4-Bit ECC Error Address Register 2	Section 3.22
D8h	NANDERRVAL1	NAND Flash 4-Bit ECC Error Value Register 1	Section 3.23
DCh	NANDERRVAL2	NAND Flash 4-Bit ECC Error Value Register 2	Section 3.24

3.1 Module ID Register (MIDR)

This is a read-only register indicating the module ID of the EMIFA. The MIDR is shown in Figure 19 and described in Table 27.

Figure 19. Module ID Register (MIDR)

31	0
REV	
R-4000 0205h	

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

Table 27. Module ID Register (MIDR) Field Descriptions

Bit	Field	Value	Description
31-0	REV	4000 0205h	Module ID of EMIFA.

3.2 Asynchronous Wait Cycle Configuration Register (AWCC)

The asynchronous wait cycle configuration register (AWCC) is used to configure the parameters for extended wait cycles. Both the polarity of the EMA_WAIT pin(s) and the maximum allowable number of extended wait cycles can be configured. The AWCC is shown in Figure 20 and described in Table 28. Not all devices support both EMA_WAIT[1] and EMA_WAIT[0], see the device-specific data manual to determine support on each device.

NOTE: The EW bit in the asynchronous *n* configuration register (CE*n*CFG) must be set to allow for the insertion of extended wait cycles.

Figure 20. Asynchronous Wait Cycle Configuration Register (AWCCR)

31	30	29	28	27	24	23	22	21	20	19	18	17	16
Reserved		WP1	WP0	Reserved		CS5_WAIT		CS4_WAIT		CS3_WAIT		CS2_WAIT	
R-0		R/W-1	R/W-1	R-0		R/W-0		R/W-0		R/W-0		R/W-0	

15	8	7	0
Reserved		MAX_EXT_WAIT	
R-0		R/W-80h	

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

Table 28. Asynchronous Wait Cycle Configuration Register (AWCCR) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29	WP1		EMA_WAIT[1] polarity bit. This bit defines the polarity of the EMA_WAIT[1] pin.
		0	Insert wait cycles if EMA_WAIT[1] pin is low.
		1	Insert wait cycles if EMA_WAIT[1] pin is high.
28	WP0		EMA_WAIT[0] polarity bit. This bit defines the polarity of the EMA_WAIT[0] pin.
		0	Insert wait cycles if EMA_WAIT[0] pin is low.
		1	Insert wait cycles if EMA_WAIT[0] pin is high.
27-24	Reserved	0	Reserved
23-22	CS5_WAIT	0-3h	Chip Select 5 WAIT signal selection. This signal determines which EMA_WAIT[*n*] signal will be used for memory accesses to chip select 5 memory space.
		0	EMA_WAIT[0] pin is used to control external wait states.
		1h	EMA_WAIT[1] pin is used to control external wait states.
		2h-3h	Reserved
21-20	CS4_WAIT	0-3h	Chip Select 4 WAIT signal selection. This signal determines which EMA_WAIT[*n*] signal will be used for memory accesses to chip select 4 memory space.
		0	EMA_WAIT[0] pin is used to control external wait states.
		1h	EMA_WAIT[1] pin is used to control external wait states.
		2h-3h	Reserved
19-18	CS3_WAIT	0-3h	Chip Select 3 WAIT signal selection. This signal determines which EMA_WAIT[*n*] signal will be used for memory accesses to chip select 3 memory space.
		0	EMA_WAIT[0] pin is used to control external wait states.
		1h	EMA_WAIT[1] pin is used to control external wait states.
		2h-3h	Reserved
17-16	CS2_WAIT	0-3h	Chip Select 2 WAIT signal selection. This signal determines which EMA_WAIT[*n*] signal will be used for memory accesses to chip select 2 memory space.
		0	EMA_WAIT[0] pin is used to control external wait states..
		1h	EMA_WAIT[1] pin is used to control external wait states.
		2h-3h	Reserved
15-8	Reserved	0	Reserved
7-0	MAX_EXT_WAIT	0-FFh	Maximum extended wait cycles. The EMIFA will wait for a maximum of (MAX_EXT_WAIT + 1) × 16 clock cycles before it stops inserting asynchronous wait cycles and proceeds to the hold period of the access.

3.3 SDRAM Configuration Register (SDCR)

The SDRAM configuration register (SDCR) 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. In addition, this register is used to put the attached SDRAM device into Self-Refresh mode. The SDCR is shown in Figure 21 and described in Table 29.

NOTE: Writing to the lower three bytes of this register will cause the EMIFA to start the SDRAM initialization sequence described in Section 2.4.4.

Figure 21. SDRAM Configuration Register (SDCR)

31	30	29	28				24
SR	PD	PDWR	Reserved				
R/W-0	R/W-0	R/W-0	R-0				
23							16
Reserved							
R-0							
15	14	13	12	11		9	8
Reserved	NM(A)	Reserved		CL			BIT11_9LOCK
R-0	R/W-0	R-0		R/W-3h			R/W-0
7	6		4	3	2		0
Reserved	IBANK			Reserved	PAGESIZE		
R-0	R/W-2h			R-0	R/W-0		

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

#IMPLIED The NM bit must be set to 1 if the EMIFA on your device only has 16 data bus pins.

Table 29. SDRAM Configuration Register (SDCR) Field Descriptions

Bit	Field	Value	Description
31	SR		Self-Refresh mode bit. This bit controls entering and exiting of the Self-Refresh mode described in Section 2.4.7. The field should be written using a byte-write to the upper byte of SDCR to avoid triggering the SDRAM initialization sequence.
		0	Writing a 0 to this bit will cause connected SDRAM devices and the EMIFA to exit the Self-Refresh mode.
		1	Writing a 1 to this bit will cause connected SDRAM devices and the EMIFA to enter the Self-Refresh mode.
30	PD		Power Down bit. This bit controls entering and exiting of the power-down mode. The field should be written using a byte-write to the upper byte of SDCR to avoid triggering the SDRAM initialization sequence. If both SR and PD bits are set, the EMIFA will go into Self Refresh.
		0	Writing a 0 to this bit will cause connected SDRAM devices and the EMIFA to exit the power-down mode.
		1	Writing a 1 to this bit will cause connected SDRAM devices and the EMIFA to enter the power-down mode.
29	PDWR		Perform refreshes during power down. Writing a 1 to this bit will cause EMIFA to exit power-down state and issue and AUTO REFRESH command every time Refresh May level is set.
28-15	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
14	NM		Narrow mode bit. This bit defines whether a 16- or 32-bit-wide SDRAM is connected to the EMIFA. This bit field must always be set to 1. Writing to this field triggers the SDRAM initialization sequence.
		0	32-bit SDRAM data bus is used.
		1	16-bit SDRAM data bus is used.
13-12	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.

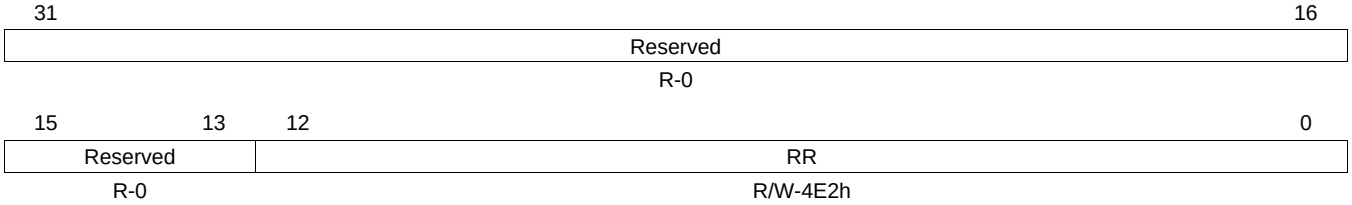
Table 29. SDRAM Configuration Register (SDCR) Field Descriptions (continued)

Bit	Field	Value	Description
11-9	CL	0-7h	CAS Latency. This field defines the CAS latency to be used when accessing connected SDRAM devices. A 1 must be simultaneously written to the BIT11_9LOCK bit field of this register in order to write to the CL bit field. Writing to this field triggers the SDRAM initialization sequence.
		0-1h	Reserved
		2h	CAS latency = 2 EMA_CLK cycles
		3h	CAS latency = 3 EMA_CLK cycles
		4h-7h	Reserved
8	BIT11_9LOCK		Bits 11 to 9 lock. CL can only be written if BIT11_9LOCK is simultaneously written with a 1. BIT11_9LOCK is always read as 0. Writing to this field triggers the SDRAM initialization sequence.
		0	CL cannot be written.
		1	CL can be written.
7	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
6-4	IBANK	0-7h	Internal SDRAM Bank size. This field defines number of banks inside the connected SDRAM devices. Writing to this field triggers the SDRAM initialization sequence.
		0	1 bank SDRAM devices.
		1	2 bank SDRAM devices.
		2	4 bank SDRAM devices.
		3h-7h	Reserved.
3	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
2-0	PAGESIZE	0-7h	Page Size. This field defines the internal page size of connected SDRAM devices. Writing to this field triggers the SDRAM initialization sequence.
		0	8 column address bits (256 elements per row)
		1h	9 column address bits (512 elements per row)
		2h	10 column address bits (1024 elements per row)
		3h	11 column address bits (2048 elements per row)
		4h-7h	Reserved

3.4 SDRAM Refresh Control Register (SDRCR)

The SDRAM refresh control register (SDRCR) is used to configure the rate at which connected SDRAM devices will be automatically refreshed by the EMIFA. Refer to Section 2.4.6 on the refresh controller for more details. The SDRCR is shown in Figure 22 and described in Table 30.

Figure 22. SDRAM Refresh Control Register (SDRCR)



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

Table 30. SDRAM Refresh Control Register (SDRCR) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
12-0	RR	0-1FFFh	Refresh Rate. This field is used to define the SDRAM refresh period in terms of EMA_CLK cycles. Writing a value < 0x0020 to this field will cause it to be loaded with (2 × T_RFC) + 1 value from the SDRAM timing register (SDTIMR).

3.5 Asynchronous *n* Configuration Registers (CE2CFG-CE5CFG)

The asynchronous *n* configuration registers (CE2CFG, CE3CFG, CE4CFG, and CE5CFG) are used to configure the shaping of the address and control signals during an access to asynchronous memory connected to CS2, CS3, CS4, and CS5, respectively. It is also used to program the width of asynchronous interface and to select from various modes of operation. This register can be written prior to any transfer, and any asynchronous transfer following the write will use the new configuration. The CE*n*CFG is shown in Figure 23 and described in Table 31.

Figure 23. Asynchronous *n* Configuration Register (CE*n*CFG)

31	30	29 26	25 24
SS	EW(A)	W_SETUP	W_STROBE(B)
R/W-0	R/W-0	R/W-Fh	R/W-3Fh

23 20	19 17	16
W_STROBE(B)	W_HOLD	R_SETUP
R/W-3Fh	R/W-7h	R/W-Fh

15 13	12 7	6 4	3 2	1 0
R_SETUP	R_STROBE(B)	R_HOLD	TA	ASIZE
R/W-Fh	R/W-3Fh	R/W-7h	R/W-3h	R/W-0

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

#IMPLIED The EW bit must be cleared to 0 when operating in NAND Flash mode.

#IMPLIED This bit field must be cleared to 0 if the EMIFA on your device does not have an EMA_WAIT pin.

Table 31. Asynchronous *n* Configuration Register (CE*n*CFG) Field Descriptions

Bit	Field	Value	Description
31	SS		Select Strobe bit. This bit defines whether the asynchronous interface operates in Normal Mode or Select Strobe Mode. See Section 2.5 for details on the two modes of operation.
		0	Normal Mode enabled.
		1	Select Strobe Mode enabled.
30	EW		Extend Wait bit. This bit defines whether extended wait cycles will be enabled. See Section 2.5.7 on extended wait cycles for details. This bit field must be set to 0 if the EMIFA on your device does not have an EMA_WAIT pin.
		0	Extended wait cycles disabled.
		1	Extended wait cycles enabled.
29-26	W_SETUP	0-Fh	Write setup width in EMA_CLK cycles, minus one cycle. See Section 2.5.3 for details.
25-20	W_STROBE	0-3Fh	Write strobe width in EMA_CLK cycles, minus one cycle. See Section 2.5.3 for details.
19-17	W_HOLD	0-7h	Write hold width in EMA_CLK cycles, minus one cycle. See Section 2.5.3 for details.
16-13	R_SETUP	0-Fh	Read setup width in EMA_CLK cycles, minus one cycle. See Section 2.5.3 for details.
12-7	R_STROBE	0-3Fh	Read strobe width in EMA_CLK cycles, minus one cycle. See Section 2.5.3 for details.
6-4	R_HOLD	0-7h	Read hold width in EMA_CLK cycles, minus one cycle. See Section 2.5.3 for details.
3-2	TA	0-3h	Minimum Turn-Around time. This field defines the minimum number of EMA_CLK cycles between reads and writes, minus one cycle. See Section 2.5.3 for details.
1-0	ASIZE	0-3h	Asynchronous Data Bus Width. This field defines the width of the asynchronous device's data bus.
		0	8-bit data bus
		1h	16-bit data bus
		2h-3h	Reserved

3.6 SDRAM Timing Register (SDTIMR)

The SDRAM timing register (SDTIMR) is used to program many of the SDRAM timing parameters. Consult the SDRAM datasheet for information on the appropriate values to program into each field. The SDTIMR is shown in Figure 24 and described in Table 32.

Figure 24. SDRAM Timing Register (SDTIMR)

31 27	26 24	23	22 20	19	18 16
T_RFC	T_RP	Rsvd	T_RCD	Rsvd	T_WR
R/W-8h	R/W-2h	R-0	R/W-2h	R-0	R/W-1h

15 12	11 8	7	6 4	3 0
T_RAS	T_RC	Rsvd	T_RRD	Reserved
R/W-5h	R/W-8h	R-0	R/W-1h	R-0

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

Table 32. SDRAM Timing Register (SDTIMR) Field Descriptions

Bit	Field	Value	Description
31-27	T_RFC	0-1Fh	Specifies the Trfc value of the SDRAM. This defines the minimum number of EMA_CLK cycles from Refresh (REFR) to Refresh (REFR), minus 1: T_RFC = (Trfc/t_{EMA_CLK}) - 1
26-24	T_RP	0-7h	Specifies the Trp value of the SDRAM. This defines the minimum number of EMA_CLK cycles from Precharge (PRE) to Activate (ACTV) or Refresh (REFR) command, minus 1: T_RP = (Trp/t_{EMA_CLK}) - 1
23	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
22-20	T_RCD	0-7h	Specifies the Trcd value of the SDRAM. This defines the minimum number of EMA_CLK cycles from Active (ACTV) to Read (READ) or Write (WRT), minus 1: T_RCD = (Trcd/t_{EMA_CLK}) - 1
19	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
18-16	T_WR	0-7h	Specifies the Twr value of the SDRAM. This defines the minimum number of EMA_CLK cycles from last Write (WRT) to Precharge (PRE), minus 1: T_WR = (Twr/t_{EMA_CLK}) - 1
15-12	T_RAS	0-Fh	Specifies the Tras value of the SDRAM. This defines the minimum number of EMA_CLK clock cycles from Activate (ACTV) to Precharge (PRE), minus 1: T_RAS = (Tras/t_{EMA_CLK}) - 1
11-8	T_RC	0-Fh	Specifies the Trc value of the SDRAM. This defines the minimum number of EMA_CLK clock cycles from Activate (ACTV) to Activate (ACTV), minus 1: T_RC = (Trc/t_{EMA_CLK}) - 1
7	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
6-4	T_RRD	0-7h	Specifies the Trrd value of the SDRAM. This defines the minimum number of EMA_CLK clock cycles from Activate (ACTV) to Activate (ACTV) for a different bank, minus 1: T_RRD = (Trrd/t_{EMA_CLK}) - 1
3-0	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.

3.7 SDRAM Self Refresh Exit Timing Register (SDSRETR)

The SDRAM self refresh exit timing register (SDSRETR) is used to program the amount of time between when the SDRAM exits Self-Refresh mode and when the EMIFA issues another command. The SDSRETR is shown in Figure 25 and described in Table 33.

Figure 25. SDRAM Self Refresh Exit Timing Register (SDSRETR)

31	16
Reserved	
R-0	

15	5	4	0
Reserved		T_XS	
R-0		R/W-9h	

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

Table 33. SDRAM Self Refresh Exit Timing Register (SDSRETR) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved. The reserved bit location is always read as 0.
4-0	T_XS	0-1Fh	This field specifies the minimum number of ECLKOUT cycles from Self-Refresh exit to any command, minus one. T_XS = Txsr / t_{EMA_CLK} - 1

3.8 EMIFA Interrupt Raw Register (INTRAW)

The EMIFA interrupt raw register (INTRAW) is used to monitor and clear the EMIFA's hardware-generated Asynchronous Timeout Interrupt. The AT bit in this register will be set when an Asynchronous Timeout occurs regardless of the status of the EMIFA interrupt mask set register (INTMSKSET) and EMIFA interrupt mask clear register (INTMSKCLR). Writing a 1 to this bit will clear it. The EMIFA on some devices does not have the EMA_WAIT pin; therefore, these registers and fields are reserved on those devices. The INTRAW is shown in Figure 26 and described in Table 34.

Figure 26. EMIFA Interrupt Raw Register (INTRAW)

31	8
Reserved	
R-0	

7	3	2	1	0
Reserved		WR	LT	AT
R-0		R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear (writing 0 has no effect); -*n* = value after reset

Table 34. EMIFA Interrupt Raw Register (INTRAW) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
2	WR		Wait Rise. This bit is set to 1 by hardware to indicate that a rising edge on the EMA_WAIT pin has occurred.
		0	Indicates that a rising edge has not occurred on the EMA_WAIT pin. Writing a 0 has no effect.
		1	Indicates that a rising edge has occurred on the EMA_WAIT pin. Writing a 1 will clear this bit and the WR_MASKED bit in the EMIFA interrupt masked register (INTMSK).
1	LT		Line Trap. Set to 1 by hardware to indicate illegal memory access type or invalid cache line size.
		0	Writing a 0 has no effect.
		1	Indicates that a line trap has occurred. Writing a 1 will clear this bit as well as the LT_MASKED bit in the EMIFA interrupt masked register(INTMSK).
0	AT		Asynchronous Timeout. This bit is set to 1 by hardware to indicate that during an extended asynchronous memory access cycle, the EMA_WAIT pin did not go inactive within the number of cycles defined by the MAX_EXT_WAIT field in the asynchronous wait cycle configuration register (AWCC).
		0	Indicates that an Asynchronous Timeout has not occurred. Writing a 0 has no effect.
		1	Indicates that an Asynchronous Timeout has occurred. Writing a 1 will clear this bit as well as the AT_MASKED bit in the EMIFA interrupt masked register (INTMSK).

3.9 EMIFA Interrupt Masked Register (INTMSK)

Like the EMIFA interrupt raw register (INTRAW), the EMIFA interrupt masked register (INTMSK) is used to monitor and clear the status of the EMIFA's hardware-generated Asynchronous Timeout Interrupt. The main difference between the two registers is that when the AT_MASKED bit in this register is set, an active-high pulse will be sent to the CPU interrupt controller. Also, the AT_MASKED bit field in INTMSK is only set to 1 if the associated interrupt has been enabled in the EMIFA interrupt mask set register (INTMSKSET). The EMIFA on some devices does not have the EMA_WAIT pin, therefore, these registers and fields are reserved on those devices. The INTMSK is shown in Figure 27 and described in Table 35.

Figure 27. EMIFA Interrupt Mask Register (INTMSK)

31	8
Reserved	
R-0	

7	3	2	1	0
Reserved		WR_MASKED	LT_MASKED	AT_MASKED
R-0		R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear (writing 0 has no effect); -*n* = value after reset

Table 35. EMIFA Interrupt Mask Register (INTMSK) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
2	WR_MASKED		Wait Rise Masked. This bit is set to 1 by hardware to indicate a rising edge has occurred on the EMA_WAIT pin, provided that the WR_MASK_SET bit is set to 1 in the EMIFA interrupt mask set register (INTMSKSET).
		0	Indicates that a wait rise interrupt has not been generated. Writing a 0 has no effect.
		1	Indicates that a wait rise interrupt has been generated. Writing a 1 will clear this bit and the WR bit in the EMIFA interrupt raw register (INTRAW).
1	LT_MASKED		Masked Line Trap. Set to 1 by hardware to indicate illegal memory access type or invalid cache line size, only if the LT_MASK_SET bit in the EMIFA interrupt mask set register (INTMSKSET) is set to 1.
		0	Writing a 0 has no effect.
		1	Writing a 1 will clear this bit as well as the LT bit in the EMIFA interrupt raw register(INTRAW).
0	AT_MASKED		Asynchronous Timeout Masked. This bit is set to 1 by hardware to indicate that during an extended asynchronous memory access cycle, the EMA_WAIT pin did not go inactive within the number of cycles defined by the MAX_EXT_WAIT field in the asynchronous wait cycle configuration register (AWCC), provided that the AT_MASK_SET bit is set to 1 in the EMIFA interrupt mask set register (INTMSKSET).
		0	Indicates that an Asynchronous Timeout Interrupt has not been generated. Writing a 0 has no effect.
		1	Indicates that an Asynchronous Timeout Interrupt has been generated. Writing a 1 will clear this bit as well as the AT bit in the EMIFA interrupt raw register (INTRAW).

3.10 EMIFA Interrupt Mask Set Register (INTMSKSET)

The EMIFA interrupt mask set register (INTMSKSET) is used to enable the Asynchronous Timeout Interrupt. If read as 1, the AT_MASKED bit in the EMIFA interrupt masked register (INTMSK) will be set and an interrupt will be generated when an Asynchronous Timeout occurs. If read as 0, the AT_MASKED bit will always read 0 and no interrupt will be generated when an Asynchronous Timeout occurs. Writing a 1 to the AT_MASK_SET bit enables the Asynchronous Timeout Interrupt. The EMIFA on some devices does not have the EMA_WAIT pin; therefore, these registers and fields are reserved on those devices. The INTMSKSET is shown in Figure 28 and described in Table 36.

Figure 28. EMIFA Interrupt Mask Set Register (INTMSKSET)

31	16
Reserved	
R-0	

15	3	2	1	0
Reserved		WR_MASK_SET	Reserved	AT_MASK_SET
R-0		R/W-0	R-0	R/W-0

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

Table 36. EMIFA Interrupt Mask Set Register (INTMSKSET) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
2	WR_MASK_SET		Wait Rise Mask Set. This bit determines whether or not the wait rise Interrupt is enabled. Writing a 1 to this bit sets this bit, sets the WR_MASK_CLR bit in the EMIFA interrupt mask clear register (INTMSKCLR), and enables the wait rise interrupt. To clear this bit, a 1 must be written to the WR_MASK_CLR bit in INTMSKCLR.
		0	Indicates that the wait rise interrupt is disabled. Writing a 0 has no effect.
		1	Indicates that the wait rise interrupt is enabled. Writing a 1 sets this bit and the WR_MASK_CLR bit in the EMIFA interrupt mask clear register (INTMSKCLR).
1	LT_MASK_SET		Mask set for LT_MASKED bit in the EMIFA interrupt mask register (INTMSK).
		0	Indicates that the line trap interrupt is disabled. Writing a 0 has no effect.
		1	Indicates that the line trap interrupt is enabled. Writing a 1 sets this bit and the LT_MASK_CLR bit in the EMIFA interrupt mask clear register (INTMSKCLR).
0	AT_MASK_SET		Asynchronous Timeout Mask Set. This bit determines whether or not the Asynchronous Timeout Interrupt is enabled. Writing a 1 to this bit sets this bit, sets the AT_MASK_CLR bit in the EMIFA interrupt mask clear register (INTMSKCLR), and enables the Asynchronous Timeout Interrupt. To clear this bit, a 1 must be written to the AT_MASK_CLR bit of the EMIFA interrupt mask clear register (INTMSKCLR).
		0	Indicates that the Asynchronous Timeout Interrupt is disabled. Writing a 0 has no effect.
		1	Indicates that the Asynchronous Timeout Interrupt is enabled. Writing a 1 sets this bit and the AT_MASK_CLR bit in the EMIFA interrupt mask clear register (INTMSKCLR).

3.11 EMIFA Interrupt Mask Clear Register (INTMSKCLR)

The EMIFA interrupt mask clear register (INTMSKCLR) is used to disable the Asynchronous Timeout Interrupt. If read as 1, the AT_MASKED bit in the EMIFA interrupt masked register (INTMSK) will be set and an interrupt will be generated when an Asynchronous Timeout occurs. If read as 0, the AT_MASKED bit will always read 0 and no interrupt will be generated when an Asynchronous Timeout occurs. Writing a 1 to the AT_MASK_CLR bit disables the Asynchronous Timeout Interrupt. The EMIFA on some devices does not have the EMA_WAIT pin, therefore, these registers and fields are reserved on those devices. The INTMSKCLR is shown in Figure 29 and described in Table 37.

Figure 29. EMIFA Interrupt Mask Clear Register (INTMSKCLR)

31	16
Reserved	
R-0	

15	3	2	1	0
Reserved		WR_MASK_CLR	Reserved	AT_MASK_CLR
R-0		R/W-0	R-0	R/W-0

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

Table 37. EMIFA Interrupt Mask Clear Register (INTMSKCLR) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0.
2	WR_MASK_CLR		Wait Rise Mask Clear. This bit determines whether or not the wait rise interrupt is enabled. Writing a 1 to this bit clears this bit, clears the WR_MASK_SET bit in the EMIFA interrupt mask set register (INTMSKSET), and disables the wait rise interrupt. To set this bit, a 1 must be written to the WR_MASK_SET bit in INTMSKSET.
		0	Indicates that the wait rise interrupt is disabled. Writing a 0 has no effect.
		1	Indicates that the wait rise interrupt is enabled. Writing a 1 clears this bit and the WR_MASK_SET bit in the EMIFA interrupt mask set register (INTMSKSET).
1	LT_MASK_CLR		Line trap Mask Clear. This bit determines whether or not the line trap interrupt is enabled. Writing a 1 to this bit clears this bit, clears the LT_MASK_SET bit in the EMIFA interrupt mask set register (INTMSKSET), and disables the line trap interrupt. To set this bit, a 1 must be written to the LT_MASK_SET bit in INTMSKSET.
		0	Indicates that the line trap interrupt is disabled. Writing a 0 has no effect.
		1	Indicates that the line trap interrupt is enabled. Writing a 1 clears this bit and the LT_MASK_SET bit in the EMIFA interrupt mask set register (INTMSKSET).
0	AT_MASK_CLR		Asynchronous Timeout Mask Clear. This bit determines whether or not the Asynchronous Timeout Interrupt is enabled. Writing a 1 to this bit clears this bit, clears the AT_MASK_SET bit in the EMIFA interrupt mask set register (INTMSKSET), and disables the Asynchronous Timeout Interrupt. To set this bit, a 1 must be written to the AT_MASK_SET bit of the EMIFA interrupt mask set register (INTMSKSET).
		0	Indicates that the Asynchronous Timeout Interrupt is disabled. Writing a 0 has no effect.
		1	Indicates that the Asynchronous Timeout Interrupt is enabled. Writing a 1 clears this bit and the AT_MASK_SET bit in the EMIFA interrupt mask set register (INTMSKSET).

3.12 NAND Flash Control Register (NANDFCR)

The NAND Flash control register (NANDFCR) is shown in Figure 30 and described in Table 38.

Figure 30. NAND Flash Control Register (NANDFCR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8
Reserved		4BITECC_ADD_CALC_START	4BITECC_START	CS5ECC	CS4ECC	CS3ECC	CS2ECC
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
Reserved		4BITECCSEL		CS5NAND	CS4NAND	CS3NAND	CS2NAND
R-0		R/W-0		R/W-0	R/W-0	R/W-0	R/W-0

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

Table 38. NAND Flash Control Register (NANDFCR) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved
13	4BITECC_ADD_CALC_START		NAND Flash 4-bit ECC address and error value calculation Start. Set to 1 to start 4_bit ECC error address and error value calculation on read syndrome. This bit is cleared when any of the NAND Flash error address registers or NAND Flash error value registers are read.
		1	start 4_bit ECC error address and error value calculation on read syndrome.
12	4BITECC_START		Nand Flash 4-bit ECC start for the selected chip select. Set to 1 to start 4_bit ECC calculation on data for NAND Flash on chip select selected by bit 4BITECCSEL. This bit is cleared when ay of the NAND Flash 4_bit ECC registers are read.
		1	start 4_bit ECC calculation on data for NAND Flash on chip select selected by bit 4BITECCSEL.
11	CS5ECC		NAND Flash ECC start for chip select 5. Set to 1 to start 1_bit ECC calculation on data for NAND Flash for this chip select. This bit is cleared when CS5 1_bit ECC register is read.
		0	Do not start ECC calculation.
		1	Start ECC calculation on data for NAND Flash on $\overline{\text{EMA\_CS5}}$.
10	CS4ECC		NAND Flash ECC start for chip select 4. Set to 1 to start 1_bit ECC calculation on data for NAND Flash for this chip select. This bit is cleared when CS4 1_bit ECC register is read.
		0	Do not start ECC calculation.
		1	Start ECC calculation on data for NAND Flash on $\overline{\text{EMA\_CS4}}$.
9	CS3ECC		NAND Flash ECC start for chip select 3. Set to 1 to start 1_bit ECC calculation on data for NAND Flash for this chip select. This bit is cleared when CS3 1_bit ECC register is read.
		0	Do not start ECC calculation.
		1	Start ECC calculation on data for NAND Flash on $\overline{\text{EMA\_CS3}}$.
8	CS2ECC		NAND Flash ECC start for chip select 2. This bit is cleared when CS2 1_bit ECC register is read.
		0	Do not start ECC calculation.
		1	Start ECC calculation on data for NAND Flash on $\overline{\text{EMA\_CS2}}$.
7-6	Reserved	0	Reserved

Table 38. NAND Flash Control Register (NANDFCR) Field Descriptions (continued)

Bit	Field	Value	Description
5-4	4BITECCSEL	0-3h	4-bit ECC selection. This field selects the chip select on which 4-bit ECC will be calculated.
		0	ECC will be calculated for CS2.
		1h	ECC will be calculated for CS3.
		2h	ECC will be calculated for CS4.
		3h	ECC will be calculated for CS5.
3	CS5NAND		NAND Flash mode for chip select 5.
		0	Not using NAND Flash.
		1	Using NAND Flash on $\overline{\text{EMA\_CS5}}$.
2	CS4NAND		NAND Flash mode for chip select 4.
		0	Not using NAND Flash.
		1	Using NAND Flash on $\overline{\text{EMA\_CS4}}$.
1	CS3NAND		NAND Flash mode for chip select 3.
		0	Not using NAND Flash.
		1	Using NAND Flash on $\overline{\text{EMA\_CS3}}$.
0	CS2NAND		NAND Flash mode for chip select 2.
		0	Not using NAND Flash.
		1	Using NAND Flash on $\overline{\text{EMA\_CS2}}$.

3.13 NAND Flash Status Register (NANDFSR)

The NAND Flash status register (NANDFSR) is shown in Figure 31 and described in Table 39.

Figure 31. NAND Flash Status Register (NANDFSR)

31	18	17	16
Reserved		ECC_ERRNUM	
R-0		R-0	

15	12	11	8	7	2	1	0
Reserved		ECC_STATE		Reserved		WAITST[*n*]	
R-0		R-0		R-0		R-0	

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

Table 39. NAND Flash Status Register (NANDFSR) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved
17-16	ECC_ERRNUM	0-3h	Number of Errors found after the 4-Bit ECC Error Address and Error Value Calculation.
		0	1 error found.
		1h	2 errors found.
		2h	3 errors found.
		3h	4 errors found.
15-12	Reserved	0	Reserved.
11-8	ECC_STATE	0-Fh	ECC correction state while performing 4-bit ECC Address and Error Value Calculation
		0	No errors detected
		1h	Errors cannot be corrected (5 or more)
		2h	Error correction complete(errors on bit 8 or 9).
		3h	Error correction complete(error exists).
		4h	Reserved.
		5h	Calculating number of errors
		6h-7h	Preparing for error search
		8h	Searching for errors
		9h-Bh	Reserved.
		Ch-Fh	Calculating error value
7-2	Reserved	0	Reserved.
1-0	WAITST[*n*]		Status of the EMA_WAIT[*n*] input pins. Not all devices support both EMA_WAIT[1] and EMA_WAIT[0], see the device-specific data manual to determine support on each device. The WP*n* bit in the asynchronous wait cycle configuration register (AWCC) has no effect on WAITST.
		0	EMA_WAIT[*n*] pin is low.
		1	EMA_WAIT[*n*] pin is high.

3.14 Page Mode Control Register (PMCR)

The page mode control register (PMCR) is shown in Figure 32 and described in Table 40. This register is configured when using NOR Flash page mode.

Figure 32. Page Mode Control Register (PMCR)

31	26	25	24
CS5_PG_DEL		CS5_PG_SIZE	CS5_PG_MD_EN
R/W-3Fh		R/W-0	R/W-0
23	18	17	16
CS4_PG_DEL		CS4_PG_SIZE	CS4_PG_MD_EN
R/W-3Fh		R/W-0	R/W-0
15	10	9	8
CS3_PG_DEL		CS3_PG_SIZE	CS3_PG_MD_EN
R/W-3Fh		R/W-0	R/W-0
7	2	1	0
CS2_PG_DEL		CS2_PG_SIZE	CS2_PG_MD_EN
R/W-3Fh		R/W-0	R/W-0

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

Table 40. Page Mode Control Register (PMCR) Field Descriptions

Bit	Field	Value	Description
31-26	CS5_PG_DEL	1-3Fh	Page access delay for NOR Flash connected on CS5. Number of EMA_CLK cycles required for the page read data to be valid, minus one cycle. This value must not be cleared to 0.
25	CS5_PG_SIZE		Page Size for NOR Flash connected on CS5.
		0	Page size is 4 words
		1	Page size is 8 words
24	CS5_PG_MD_EN		Page Mode enable for NOR Flash connected on CS5.
		0	Page mode disabled for this chip select
		1	Page mode enabled for this chip select
23-18	CS4_PG_DEL	1-3Fh	Page access delay for NOR Flash connected on CS4. Number of EMA_CLK cycles required for the page read data to be valid, minus one cycle. This value must not be cleared to 0.
17	CS4_PG_SIZE		Page Size for NOR Flash connected on CS4.
		0	Page size is 4 words
		1	Page size is 8 words
16	CS4_PG_MD_EN		Page Mode enable for NOR Flash connected on CS4.
		0	Page mode disabled for this chip select
		1	Page mode enabled for this chip select
15-10	CS3_PG_DEL	1-3Fh	Page access delay for NOR Flash connected on CS3. Number of EMA_CLK cycles required for the page read data to be valid, minus one cycle. This value must not be cleared to 0.
9	CS3_PG_SIZE		Page Size for NOR Flash connected on CS3.
		0	Page size is 4 words
		1	Page size is 8 words
8	CS3_PG_MD_EN		Page Mode enable for NOR Flash connected on CS3.
		0	Page mode disabled for this chip select
		1	Page mode enabled for this chip select
7-2	CS2_PG_DEL	1-3Fh	Page access delay for NOR Flash connected on CS2. Number of EMA_CLK cycles required for the page read data to be valid, minus one cycle. This value must not be cleared to 0.
1	CS2_PG_SIZE		Page Size for NOR Flash connected on CS2.
		0	Page size is 4 words
		1	Page size is 8 words

Table 40. Page Mode Control Register (PMCR) Field Descriptions (continued)

Bit	Field	Value	Description
0	CS2_PG_MD_EN		Page Mode enable for NOR Flash connected on CS2.
		0	Page mode disabled for this chip select
		1	Page mode enabled for this chip select

3.15 NAND Flash *n* ECC Registers (NANDF1ECC-NANDF4ECC)

The NAND Flash *n* ECC register (NANDF*n*ECC) is shown in Figure 33 and described in Table 41. For 8-bit NAND Flash, the P1 to P4 bits are column parities; the P8 to P2048 bits are row parities. For 16-bit NAND Flash, the P1 through P8 bits are column parities; the P16 to P2048 bits are row parities.

Figure 33. NAND Flash *n* ECC Register (NANDF*n*ECC)

31			28	27	26	25	24
Reserved				P2048O	P1024O	P512O	P256O
R-0				R-0	R-0	R-0	R-0
23	22	21	20	19	18	17	16
P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15			12	11	10	9	8
Reserved				P2048E	P1024E	P512E	P256E
R-0				R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

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

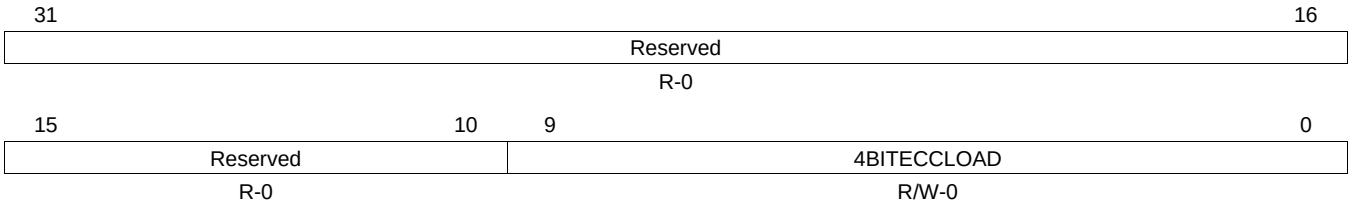
Table 41. NAND Flash *n* ECC Register (NANDF*n*ECC) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27	P2048O	0-1	ECC code calculated while reading/writing NAND Flash.
26	P1024O	0-1	ECC code calculated while reading/writing NAND Flash.
25	P512O	0-1	ECC code calculated while reading/writing NAND Flash.
24	P256O	0-1	ECC code calculated while reading/writing NAND Flash.
23	P128O	0-1	ECC code calculated while reading/writing NAND Flash.
22	P64O	0-1	ECC code calculated while reading/writing NAND Flash.
21	P32O	0-1	ECC code calculated while reading/writing NAND Flash.
20	P16O	0-1	ECC code calculated while reading/writing NAND Flash.
19	P8O	0-1	ECC code calculated while reading/writing NAND Flash.
18	P4O	0-1	ECC code calculated while reading/writing NAND Flash.
17	P2O	0-1	ECC code calculated while reading/writing NAND Flash.
16	P1O	0-1	ECC code calculated while reading/writing NAND Flash.
15-12	Reserved	0	Reserved
11	P2948E	0-1	ECC code calculated while reading/writing NAND Flash.
10	P102E	0-1	ECC code calculated while reading/writing NAND Flash.
9	P512E	0-1	ECC code calculated while reading/writing NAND Flash.
8	P256E	0-1	ECC code calculated while reading/writing NAND Flash.
7	P128E	0-1	ECC code calculated while reading/writing NAND Flash.
6	P64E	0-1	ECC code calculated while reading/writing NAND Flash.
5	P32E	0-1	ECC code calculated while reading/writing NAND Flash.
4	P15E	0-1	ECC code calculated while reading/writing NAND Flash.
3	P8E	0-1	ECC code calculated while reading/writing NAND Flash.
2	P4E	0-1	ECC code calculated while reading/writing NAND Flash.
1	P2E	0-1	ECC code calculated while reading/writing NAND Flash.
0	P1E	0-1	ECC code calculated while reading/writing NAND Flash.

3.16 NAND Flash 4-Bit ECC LOAD Register (NAND4BITECCLOAD)

The NAND Flash 4-bit ECC load register (NAND4BITECCLOAD) is shown in Figure 34 and described in Table 42.

Figure 34. NAND Flash 4-Bit ECC LOAD Register (NAND4BITECCLOAD)



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

Table 42. NAND Flash 4-Bit ECC LOAD Register (NAND4BITECCLOAD) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	4BITECCLOAD	0-3FFh	4-bit ECC load. This value is used to load the ECC values when performing the Syndrome calculation during reads.

3.17 NAND Flash 4-Bit ECC Register 1 (NAND4BITECC1)

The NAND Flash 4-bit ECC register 1 (NAND4BITECC1) is shown in Figure 35 and described in Table 43.

Figure 35. NAND Flash 4-Bit ECC Register 1 (NAND4BITECC1)

31	26	25	16
Reserved		4BITECCVAL2	
R-0		R/W-0	

15	10	9	0
Reserved		4BITECCVAL1	
R-0		R/W-0	

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

Table 43. NAND Flash 4-Bit ECC Register 1 (NAND4BITECC1) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	4BITECCVAL2	0-3FFh	Calculated 4-bit ECC or Syndrom Value2.
15-10	Reserved	0	Reserved
9-0	4BITECCVAL1	0-3FFh	Calculated 4-bit ECC or Syndrom Value1.

3.18 NAND Flash 4-Bit ECC Register 2 (NAND4BITECC2)

The NAND Flash 4-bit ECC register 2 (NAND4BITECC2) is shown in Figure 36 and described in Table 44.

Figure 36. NAND Flash 4-Bit ECC Register 2 (NAND4BITECC2)

31	26	25	16
Reserved		4BITECCVAL4	
R-0		R/W-0	

15	10	9	0
Reserved		4BITECCVAL3	
R-0		R/W-0	

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

Table 44. NAND Flash 4-Bit ECC Register 2 (NAND4BITECC2) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	4BITECCVAL4	0-3FFh	Calculated 4-bit ECC or Syndrom Value4.
15-10	Reserved	0	Reserved
9-0	4BITECCVAL3	0-3FFh	Calculated 4-bit ECC or Syndrom Value3.

3.19 NAND Flash 4-Bit ECC Register 3 (NAND4BITECC3)

The NAND Flash 4-bit ECC register 3 (NAND4BITECC3) is shown in Figure 37 and described in Table 45.

Figure 37. NAND Flash 4-Bit ECC Register 3 (NAND4BITECC3)

31	26	25	16
Reserved		4BITECCVAL6	
R-0		R/W-0	

15	10	9	0
Reserved		4BITECCVAL5	
R-0		R/W-0	

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

Table 45. NAND Flash 4-Bit ECC Register 3 (NAND4BITECC3) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	4BITECCVAL6	0-3FFh	Calculated 4-bit ECC or Syndrom Value6.
15-10	Reserved	0	Reserved
9-0	4BITECCVAL5	0-3FFh	Calculated 4-bit ECC or Syndrom Value5.

3.20 NAND Flash 4-Bit ECC Register 4 (NAND4BITECC4)

The NAND Flash 4-bit ECC register 4 (NAND4BITECC4) is shown in Figure 38 and described in Table 46.

Figure 38. NAND Flash 4-Bit ECC Register 4 (NAND4BITECC4)

31	26	25	16
Reserved		4BITECCVAL8	
R-0		R/W-0	

15	10	9	0
Reserved		4BITECCVAL7	
R-0		R/W-0	

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

Table 46. NAND Flash 4-Bit ECC Register 4 (NAND4BITECC4) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	4BITECCVAL8	0-3FFh	Calculated 4-bit ECC or Syndrom Value8.
15-10	Reserved	0	Reserved
9-0	4BITECCVAL7	0-3FFh	Calculated 4-bit ECC or Syndrom Value7.

3.21 NAND Flash 4-Bit ECC Error Address Register 1 (NANDERRADD1)

The NAND Flash 4-bit ECC error register 1 (NANDERRADD1) is shown in Figure 39 and described in Table 47.

Figure 39. NAND Flash 4-Bit ECC Error Address Register 1 (NANDERRADD1)

31	26	25	16
Reserved		4BITECCERRADD2	
R-0		R/W-0	

15	10	9	0
Reserved		4BITECCERRADD1	
R-0		R/W-0	

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

Table 47. NAND Flash 4-Bit ECC Error Address Register 1 (NANDERRADD1) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	4BITECCERRADD2	0-3FFh	Calculated 4-bit ECC Error Address 2.
15-10	Reserved	0	Reserved
9-0	4BITECCERRADD1	0-3FFh	Calculated 4-bit ECC Error Address 1.

3.22 NAND Flash 4-Bit ECC Error Address Register 2 (NANDERRADD2)

The NAND Flash 4-bit ECC error register 2 (NANDERRADD2) is shown in Figure 40and described in Table 48.

Figure 40. NAND Flash 4-Bit ECC Error Address Register 2 (NANDERRADD2)

31	26	25	16
Reserved		4BITECCERRADD4	
R-0		R/W-0	

15	10	9	0
Reserved		4BITECCERRADD3	
R-0		R/W-0	

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

Table 48. NAND Flash 4-Bit ECC Error Address Register 2 (NANDERRADD2) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	4BITECCERRADD4	0-3FFh	Calculated 4-bit ECC Error Address 4.
15-10	Reserved	0	Reserved
9-0	4BITECCERRADD3	0-3FFh	Calculated 4-bit ECC Error Address 3.

3.23 NAND Flash 4-Bit ECC Error Value Register 1 (NANDERRVAL1)

The NAND Flash 4-bit ECC error value register 1 (NANDERRVAL1) is shown in Figure 41 and described in Table 49.

Figure 41. NAND Flash 4-Bit ECC Error Value Register 1 (NANDERRVAL1)

31	26	25	16
Reserved		4BITECCERRVAL2	
R-0		R/W-0	

15	10	9	0
Reserved		4BITECCERRVAL1	
R-0		R/W-0	

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

Table 49. NAND Flash 4-Bit ECC Error Value Register 1 (NANDERRVAL1) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	4BITECCERRVAL2	0-3FFh	Calculated 4-bit ECC Error Value 2.
15-10	Reserved	0	Reserved
9-0	4BITECCERRVAL1	0-3FFh	Calculated 4-bit ECC Error Value 1.

3.24 NAND Flash 4-Bit ECC Error Value Register 2 (NANDERRVAL2)

The NAND Flash 4-bit ECC error value register 2 (NANDERRVAL2) is shown in Figure 42 and described in Table 50.

Figure 42. NAND Flash 4-Bit ECC Error Value Register 2 (NANDERRVAL2)

31	26	25	16
Reserved		4BITECCERRVAL4	
R-0		R/W-0	

15	10	9	0
Reserved		4BITECCERRVAL3	
R-0		R/W-0	

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

Table 50. NAND Flash 4-Bit ECC Error Value Register 2 (NANDERRVAL2) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25-16	4BITECCERRVAL4	0-3FFh	Calculated 4-bit ECC Error Value 4.
15-10	Reserved	0	Reserved
9-0	4BITECCERRVAL3	0-3FFh	Calculated 4-bit ECC Error Value 3.

Appendix A Example Configuration

This appendix presents an example of interfacing the EMIFA to both an SDR SDRAM device and an asynchronous flash device.

A.1 Hardware Interface

Figure 43 shows the hardware interface between the EMIFA, a Samsung K4S641632H-TC(L)70 64Mb SDRAM device, and two SHARP LH28F800BJE-PTTL90 8Mb Flash memory. The connection between the EMIFA and the SDRAM is straightforward, but the connection between the EMIFA and the flash deserves a detailed look.

The address inputs for the flash are provided by three sources. The A[12:0] address inputs are provided by a combination of the EMA_A and EMA_BA pins according to Section 2.5.1. The upper address inputs A[18:13] are provided by GPIO pins. The six GPIO pins are connected to the upper address bits of the flash memory and attached to pulldown resistors so that their value is 0 after reset and before configuring the pins as GPIO. This is necessary if the ROM bootloader is copying the secondary bootloader from the flash. More details on using GPIO pins as upper address pins can be found in Section 2.5.2. RD/$\overline{BY}$ signal from one flash is connected to EMA_WAIT pin of EMIFA. A GPIO pin can be made use of to receive the RD/$\overline{BY}$ signal coming from the second flash, as shown in Figure 43

Finally, this example configuration connects the $\overline{EMA\_WE}$ pin to the $\overline{WE}$ input of the flash and operates the EMIFA in Select Strobe Mode.

A.2 Software Configuration

The following sections describe how to configure the EMIFA registers and bit fields to interface the EMIFA with the Samsung K4S641632H-TC(L)70 SDRAM and the SHARP LH28F800BJE-PTTL90 8Mb Flash memory.

A.2.1 Configuring the SDRAM Interface

This section describes how to configure the EMIFA to interface with the Samsung K4S641632H-TC(L)70 SDRAM with a clock frequency of f_{EMA_CLK} = 100 MHz. Procedure A described in Section 2.4.5 is followed which assumes that the SDRAM power-up timing constraint were met during the SDRAM Auto-Initialization sequence after Reset.

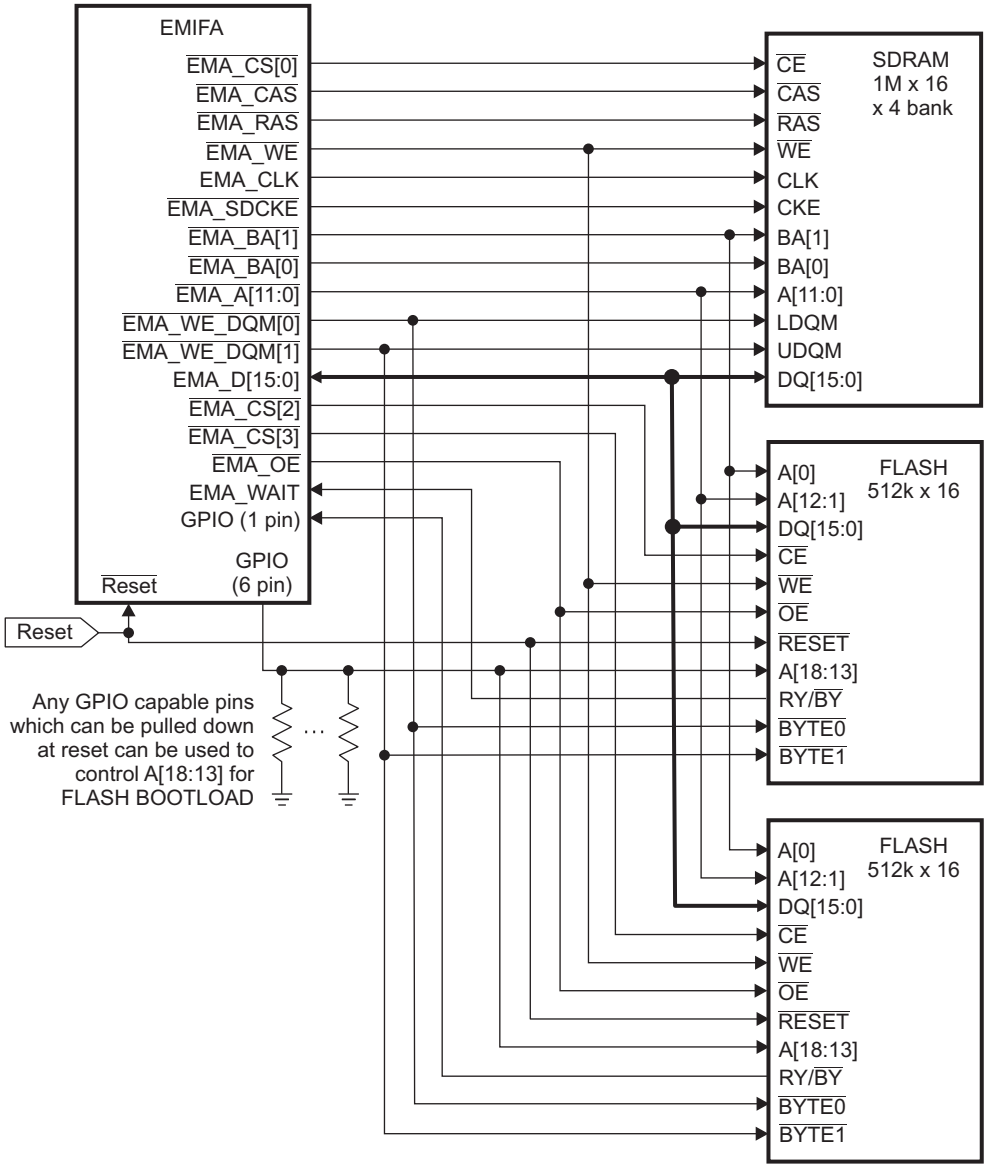
A.2.1.1 PLL Programming for the EMIFA to K4S641632H-TC(L)70 Interface

The device PLL Controller should first be programmed to select the desired EMA_CLK frequency. Before doing this, the SDRAM should be placed in Self-Refresh Mode by setting the SR bit in the SDRAM configuration register (SDCR). The SR bit should be set using a byte-write to the upper byte of the SDCR to avoid triggering the SDRAM Initialization Sequence. The EMA_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 SR bit in SDCR, again with a byte-write.

Table 51. SR Field Value For the EMIFA to K4S641632H-TC(L)70 Interface

Field	Value	Purpose
SR	1 then 0	To place the EMIFA into the self refresh state

Figure 43. Example Configuration Interface



A.2.1.2 SDRAM Timing Register (SDTIMR) Settings for the EMIFA to K4S641632H-TC(L)70 Interface

The fields of the SDRAM timing register (SDTIMR) should be programmed first as described in Table 52 to satisfy the required timing parameters for the K4S641632H-TC(L)70. Based on these calculations, a value of 6111 4610h should be written to SDTIMR. Figure 44 shows a graphical description of how SDTIMR should be programmed.

Table 52. SDTIMR Field Calculations for the EMIFA to K4S641632H-TC(L)70 Interface

Field Name	Formula	Value from K4S641632H-TC(L)70 Datasheet	Value Calculated for Field
T_RFC	T_RFC >= (t_{RFC} × f_{EMA_CLK}) - 1	t_{RC} = 68 ns (min)(1)	6
T_RP	T_RP >= (t_{RP} × f_{EMA_CLK}) - 1	t_{RP} = 20 ns (min)	1
T_RCD	T_RCD >= (t_{RCD} × f_{EMA_CLK}) - 1	t_{RCD} = 20 ns (min)	1
T_WR	T_WR >= (t_{WR} × f_{EMA_CLK}) - 1	t_{RDL} = 2 CLK = 20 ns (min)(2)	1
T_RAS	T_RAS >= (t_{RAS} × f_{EMA_CLK}) - 1	t_{RAS} = 49 ns (min)	4
T_RC	T_RC >= (t_{RC} × f_{EMA_CLK}) - 1	t_{RC} = 68 ns (min)	6
T_RRD	T_RRD >= (t_{RRD} × f_{EMA_CLK}) - 1	t_{RRD} = 14 ns (min)	1

(1) The Samsung datasheet does not specify a t_{RFC} value. Instead, Samsung specifies t_{RC} as the minimum auto refresh period.
(2) The Samsung datasheet does not specify a t_{WR} value. Instead, Samsung specifies t_{RDL} as last data in to row precharge minimum delay.

Figure 44. SDRAM Timing Register (SDTIMR)

31 27	26 24	23	22 20	19	18 16
0 0110	001	0	001	0	001
T_RFC	T_RP	Rsvd	T_RCD	Rsvd	T_WR

15 12	11 8	7	6 4	3 0
0100	0110	0	001	0000
T_RAS	T_RC	Rsvd	T_RRD	Reserved

A.2.1.3 SDRAM Self Refresh Exit Timing Register (SDSRETR) Settings for the EMIFA to K4S641632H-TC(L)70 Interface

The SDRAM self refresh exit timing register (SDSRETR) should be programmed second to satisfy the t_{XSR} timing requirement from the K4S641632H-TC(L)70 datasheet. Table 53 shows the calculation of the proper value to program into the T_XS field of this register. Based on this calculation, a value of 6h should be written to SDSRETR. Figure 45 shows how SDSRETR should be programmed.

Table 53. RR Calculation for the EMIFA to K4S641632H-TC(L)70 Interface

Field Name	Formula	Value from K4S641632H-TC(L)70 Datasheet	Value Calculated for Field
T_XS	T_XS >= (t_{XSR} × f_{EMA_CLK}) - 1	t_{RC} = 68 ns (min)[1]	6

[1] The Samsung datasheet does not specify a t_{XSR} value. Instead, Samsung specifies t_{RC} as the minimum required time after CKE going high to complete self refresh exit.

Figure 45. SDRAM Self Refresh Exit Timing Register (SDSRETR)

31	16
0000 0000 0000 0000	
Reserved	

15	5	4	0
000 0000 0000		0 0110	
Reserved		T_XS	

A.2.1.4 SDRAM Refresh Control Register (SDRCR) Settings for the EMIFA to K4S641632H-TC(L)70 Interface

The SDRAM refresh control register (SDRCR) should next be programmed to satisfy the required refresh rate of the K4S641632H-TC(L)70. Table 54 shows the calculation of the proper value to program into the RR field of this register. Based on this calculation, a value of 61Ah should be written to SDRCR. Figure 46 shows how SDRCR should be programmed.

Table 54. RR Calculation for the EMIF to K4S641632H-TC(L)70 Interface

Field Name	Formula	Values	Value Calculated for Field
RR	RR ≤ f_{EMA_CLK} × $t_{Refresh\ Period}$ / n_{cycles}	From SDRAM datasheet: $t_{Refresh\ Period}$ = 64 ms; n_{cycles} = 4096 EMIFA clock rate: f_{EMA_CLK} = 100 MHz	RR = 1562 cycles = 61Ah cycles

Figure 46. SDRAM Refresh Control Register (SDRCR)

31	19	18	16
0 0000 0000 0000		000	
Reserved		Reserved	

15	13	12	0
000		0 0110 0001 1010 (61Ah)	
Reserved		RR	

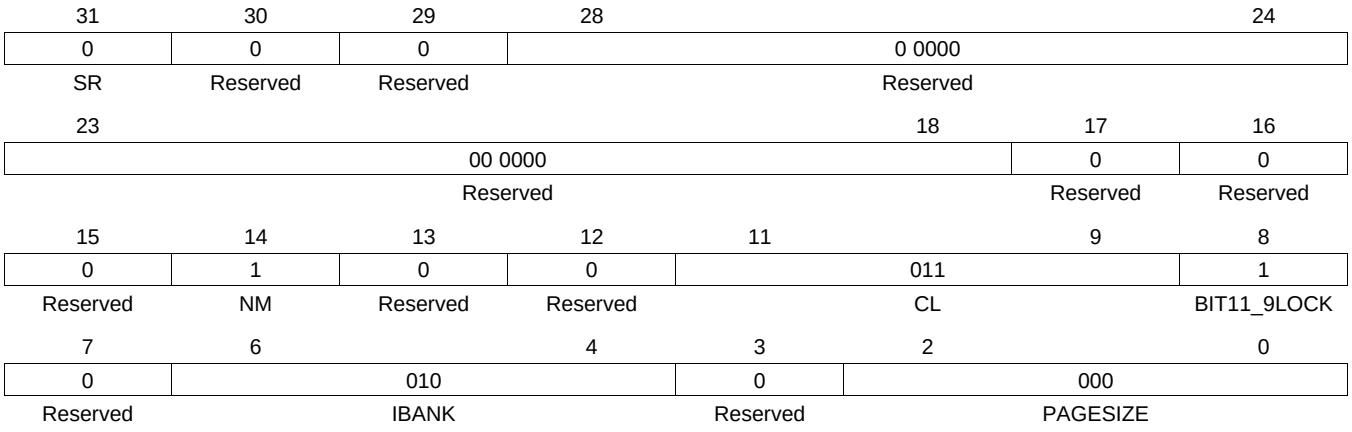
A.2.1.5 SDRAM Configuration Register (SDCR) Settings for the EMIFA to K4S641632H-TC(L)70 Interface

Finally, the fields of the SDRAM configuration register (SDCR) should be programmed as described in Table 51 to properly interface with the K4S641632H-TC(L)70 device. Based on these settings, a value of 4720h should be written to SDCR. Figure 47 shows how SDCR should be programmed. The EMIFA is now ready to perform read and write accesses to the SDRAM.

Table 55. SDCR Field Values For the EMIFA to K4S641632H-TC(L)70 Interface

Field	Value	Purpose
SR	0	To avoid placing the EMIFA into the self refresh state
NM	1	To configure the EMIFA for a 16-bit data bus
CL	011b	To select a CAS latency of 3
BIT11_9LOCK	1	To allow the CL field to be written
IBANK	010b	To select 4 internal SDRAM banks
PAGESIZE	0	To select a page size of 256 words

Figure 47. SDRAM Configuration Register (SDCR)



A.2.2 Configuring the Flash Interface

This section describes how to configure the EMIFA to interface with the two of SHARP LH28F800BJE-PTTL90 8Mb Flash memory with a clock frequency of f_{EMA_CLK} = 100 MHz. The example assumes that one flash is connected to $\overline{EMA\_CS2}$ and the other to $\overline{EMA\_CS3}$.

A.2.2.1 Asynchronous 1 Configuration Register (CE2CFG) Settings for the EMIFA to LH28F800BJE-PTTL90 Interface

The asynchronous 1 configuration register (CE2CFG) and asynchronous 2 configuration register (CE3CFG) are the only registers that is necessary to program for this asynchronous interface (assuming that one Flash is connected to $\overline{EMA\_CS2}$ and the other to $\overline{EMA\_CS3}$). The SS bit (in both registers) should be set to 1 to enable Select Strobe Mode and the ASIZE field (in both registers) should be set to 1 to select a 16-bit interface. The other fields in this register control the shaping of the EMIFA signals, and the proper values can be determined by referring to the AC Characteristics in the Flash datasheet and the device Data Manual. Based on the following calculations, a value of 8862 25BDh should be written to CE2CFG. Table 56 and Table 57 show the pertinent AC Characteristics for reads and writes to the Flash device, and Figure 48 and Figure 49 show the associated timing waveforms. Finally, Figure 50 shows programming the CE*n*CFG (*n* = 2, 3) with the calculated values.

Table 56. AC Characteristics for a Read Access

AC Characteristic	Device	Definition	Min	Max	Unit
t_{SU}	EMIFA	Setup time, read EMA_D before EMA_CLK high	6.5		ns
t_H	EMIFA	Data hold time, read EMA_D after EMA_CLK high	1		ns
t_D	EMIFA	Output delay time, EMA_CLK high to output signal valid		7	ns
t_{ELQV}	Flash	$\overline{CE}$ to Output Delay		90	ns
t_{EHQZ}	Flash	$\overline{CE}$ High to Output in High Impedance		55	ns

Table 57. AC Characteristics for a Write Access

AC Characteristic	Device	Definition	Min	Max	Unit
t_{AVAV}	Flash	Write Cycle Time	90		ns
t_{ELEH}	Flash	$\overline{CE}$ Pulse Width Low	50		ns
t_{EHEL}	Flash	$\overline{CE}$ Pulse Width High (not shown in Figure 49)	30		ns

Figure 48. LH28F800BJE-PTTL90 to EMIFA Read Timing Waveforms

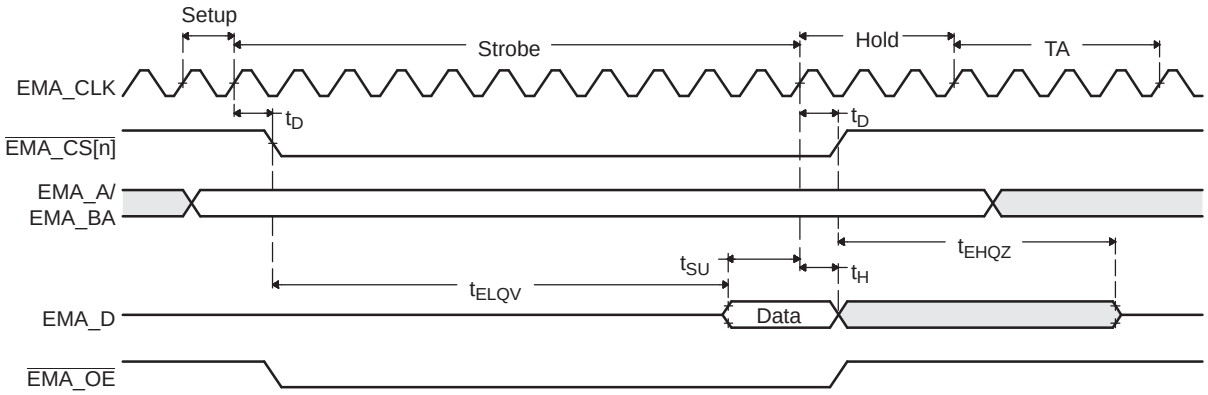
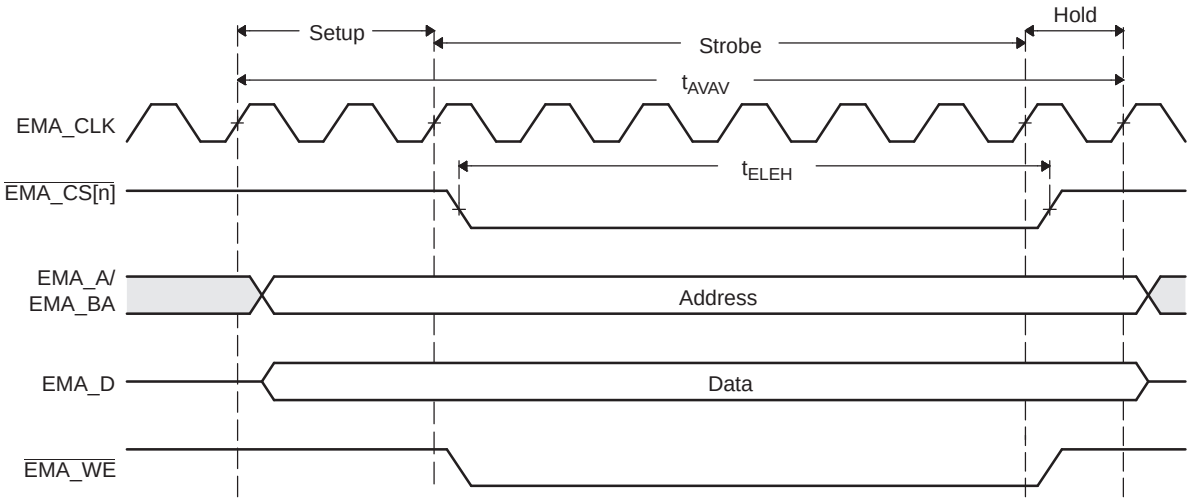


Figure 49. LH28F800BJE-PTTL90 to EMIFA Write Timing Waveforms



The R_STROBE field should be set to meet the following equation:

$R\_STROBE >= (t_D + t_{ELQV} + t_{SU}) \times f_{EMA\_CLK} - 1$

R_STROBE >= (7 ns + 90 ns + 6.5 ns) × 100 MHz - 1

R_STROBE >= 9.35

R_STROBE = 10

The R_HOLD field must be large enough to satisfy the EMIFA Data hold time, t_H:

$R\_HOLD >= t_H \times f_{EMA\_CLK} - 1$

R_HOLD >= 1 ns × 100 MHz - 1

R_HOLD >= -0.9

The R_HOLD field must also combine with the TA field to satisfy the Flash's $\overline{CE}$ High to Output in High Impedance time, t_{EHQZ}:

$R\_HOLD + TA >= (t_D + t_{EHQZ}) \times f_{EMA\_CLK} - 2$

R_HOLD + TA >= (7 ns + 55 ns) × 100 MHz - 2

R_HOLD + TA >= 4.2

The largest value that can be programmed into the TA field is 3h, therefore the following values can be used:

R_HOLD = 2

TA = 3

For Writes, the W_STROBE field should be set to satisfy the Flash's $\overline{CE}$ Pulse Width constraint, t_{ELEH}:

$W\_STROBE >= t_{ELEH} \times f_{EMA\_CLK} - 1$

W_STROBE >= 50 ns × 100 MHz - 1

W_STROBE >= 4

The W_SETUP and W_HOLD fields should combine to satisfy the Flash's $\overline{CE}$ Pulse Width High constraint, t_{EHEL}, when performing back-to-back writes:

W_SETUP + W_HOLD > = t_{EHEL} × f_{EMA_CLK} - 2

W_SETUP + W_HOLD > = 30 ns × 100 MHz - 2

W_SETUP + W_HOLD > = 1

In addition, the entire Write access length must satisfy the Flash's minimum Write Cycle Time, t_{AVAV}:

W_SETUP + W_STROBE + W_HOLD >= t_{AVAV} × f_{EMA_CLK} - 3

W_SETUP + W_STROBE + W_HOLD >= 90 ns × 100 MHz - 3

W_SETUP + W_STROBE + W_HOLD >= 6

Solving the above equations for the Write fields results in the following possible solution:

W_SETUP = 1

W_STROBE = 5

W_HOLD = 0

Adding a 10 ns (1 cycle) margin to each of the periods (excluding TA which is already at its maximum) in this example produces the following recommended values:

W_SETUP = 2h

W_STROBE = 6h

W_HOLD = 1h

R_SETUP = 1h

R_STROBE = Bh

R_HOLD = 3h

TA = 3h

Figure 50. Asynchronous *m* Configuration Register(*m*=1,2) (CE*n*CFG(*n*=2,3))

31	30	29 – 26	25 – 24
1	0	0010	00
SS	EW	W_SETUP	W_STROBE

23 – 20	19 – 17	16
0110	001	0
W_STROBE	W_HOLD	R_SETUP

15 – 13	12 – 7	6 – 4	3 – 2	1 – 0
001	001011	011	11	01
R_SETUP	R_STROBE	R_HOLD	TA	ASIZE

Appendix B Revision History

Table 58 lists the changes made since the previous version of this document.

Table 58. Document Revision History

Reference	Additions/Modifications/Deletions
Section 2.14	Changed fifth sentence in second paragraph.
Section 2.14.3	Changed first sentence.

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	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

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