

TMS320DM357 DMSoC Asynchronous External Memory Interface (EMIF)

Reference Guide



Literature Number: SPRUG33

November 2008

| | |
|---|-----------|
| Preface | 6 |
| 1 Introduction | 9 |
| 1.1 Purpose of the Peripheral | 9 |
| 1.2 Features | 9 |
| 1.3 Functional Block Diagram | 10 |
| 2 Peripheral Architecture | 10 |
| 2.1 Clock Control | 10 |
| 2.2 EMIF Requests | 10 |
| 2.3 Signal Descriptions | 11 |
| 2.4 Pin Multiplexing | 11 |
| 2.5 Asynchronous Controller and Interface | 11 |
| 3 Example Configuration | 31 |
| 3.1 Interfacing to Asynchronous SRAM (ASRAM)..... | 31 |
| 3.2 Interfacing to NAND Flash..... | 40 |
| 4 Registers | 49 |
| 4.1 Asynchronous Wait Cycle Configuration Register (AWCCR)..... | 50 |
| 4.2 Asynchronous n Configuration Registers (A1CR-A4CR)..... | 51 |
| 4.3 EMIF Interrupt Raw Register (EIRR)..... | 52 |
| 4.4 EMIF Interrupt Mask Register (EIMR) | 53 |
| 4.5 EMIF Interrupt Mask Set Register (EIMSR)..... | 54 |
| 4.6 EMIF Interrupt Mask Clear Register (EIMCR) | 55 |
| 4.7 NAND Flash Control Register (NANDFCR)..... | 56 |
| 4.8 NAND Flash Status Register (NANDFSR) | 57 |
| 4.9 NAND Flash n ECC Registers (NANDF1ECC-NANDF4ECC)..... | 57 |

List of Figures

| | | |
|----|---|----|
| 1 | EMIF Functional Block Diagram..... | 10 |
| 2 | EMIF Asynchronous Interface | 12 |
| 3 | EMIF to 8-bit and 16-bit Memory Interfaces | 12 |
| 4 | Timing Waveform of an Asynchronous Read Cycle in Normal Mode | 16 |
| 5 | Timing Waveform of an Asynchronous Write Cycle in Normal Mode | 18 |
| 6 | Timing Waveform of an Asynchronous Read Cycle in Select Strobe Mode | 20 |
| 7 | Timing Waveform of an Asynchronous Write Cycle in Select Strobe Mode..... | 22 |
| 8 | EMIF to NAND Flash Interface | 24 |
| 9 | ECC Value for 8-Bit NAND Flash | 26 |
| 10 | EMIF to 16-Bit Multiplexed HPI16 Interface | 27 |
| 11 | Connecting the EMIF to the TC55V16100FT-12 | 31 |
| 12 | Timing Waveform of an ASRAM Read | 33 |
| 13 | Timing Waveform of an ASRAM Write | 34 |
| 14 | Timing Waveform of an ASRAM Read with PCB Delays | 36 |
| 15 | Timing Waveform of an ASRAM Write with PCB Delays | 37 |
| 16 | Timing Waveform of a NAND Flash Read | 42 |
| 17 | Timing Waveform of a NAND Flash Command Write | 44 |
| 18 | Timing Waveform of a NAND Flash Address Write | 44 |
| 19 | Timing Waveform of a NAND Flash Data Write | 45 |
| 20 | Asynchronous Wait Cycle Configuration Register (AWCCR) | 50 |
| 21 | Asynchronous <i>n</i> Configuration Register (<i>An</i> CR) | 51 |
| 22 | EMIF Interrupt Raw Register (EIRR) | 52 |
| 23 | EMIF Interrupt Mask Register (EIMR) | 53 |
| 24 | EMIF Interrupt Mask Set Register (EIMSR) | 54 |
| 25 | EMIF Interrupt Mask Clear Register (EIMCR)..... | 55 |
| 26 | NAND Flash Control Register (NANDFCR)..... | 56 |
| 27 | NAND Flash Status Register (NANDFSR)..... | 57 |
| 28 | NAND Flash <i>n</i> ECC Register (NANDF <i>n</i> ECC) | 57 |

List of Tables

| | | |
|----|---|----|
| 1 | EMIF Pins | 11 |
| 2 | Behavior of $\overline{EM_CS}$ Signal Between Normal Mode and Select Strobe Mode | 11 |
| 3 | Description of the Asynchronous Configuration Register (<i>AnCR</i>) | 13 |
| 4 | Description of the Asynchronous Wait Cycle Configuration Register (AWCCR) | 14 |
| 5 | Description of the EMIF Interrupt Mask Set Register (EIMSR) | 14 |
| 6 | Description of the EMIF Interrupt Mast Clear Register (EIMCR) | 14 |
| 7 | Asynchronous Read Operation in Normal Mode | 15 |
| 8 | Asynchronous Write Operation in Normal Mode | 17 |
| 9 | Asynchronous Read Operation in Select Strobe Mode | 19 |
| 10 | Asynchronous Write Operation in Select Strobe Mode | 21 |
| 11 | Description of the NAND Flash Control Register (NANDFCR)..... | 23 |
| 12 | Configuration For NAND Flash | 23 |
| 13 | Interrupt Monitor and Control Bit Fields | 29 |
| 14 | EMIF Input Timing Requirements | 32 |
| 15 | ASRAM Output Timing Characteristics..... | 32 |
| 16 | ASRAM Input Timing Requirement for a Read..... | 32 |
| 17 | ASRAM Input Timing Requirements for a Write | 33 |
| 18 | ASRAM Timing Requirements With PCB Delays | 35 |
| 19 | EMIF Timing Requirements for TC5516100FT-12 Example | 38 |
| 20 | ASRAM Timing Requirements for TC5516100FT-12 Example | 38 |
| 21 | Measured PCB Delays for TC5516100FT-12 Example | 38 |
| 22 | Configuring A2CR for TC5516100FT-12 Example..... | 40 |
| 23 | Recommended Margins | 40 |
| 24 | EMIF Read Timing Requirements..... | 41 |
| 25 | NAND Flash Read Timing Requirements | 41 |
| 26 | NAND Flash Write Timing Requirements | 43 |
| 27 | EMIF Timing Requirements for HY27UA081G1M Example | 46 |
| 28 | NAND Flash Timing Requirements for HY27UA081G1M Example | 46 |
| 29 | Configuring A1CR for HY27UA081G1M Example | 48 |
| 30 | Configuring NANDFCR for HY27UA081G1M Example | 48 |
| 31 | External Memory Interface (EMIF) Registers..... | 49 |
| 32 | Asynchronous Wait Cycle Configuration Register (AWCCR) Field Descriptions | 50 |
| 33 | Asynchronous <i>n</i> Configuration Register (<i>AnCR</i>) Field Descriptions..... | 51 |
| 34 | EMIF Interrupt Raw Register (EIRR) Field Descriptions | 52 |
| 35 | EMIF Interrupt Mask Register (EIMR) Field Descriptions..... | 53 |
| 36 | EMIF Interrupt Mask Set Register (EIMSR) Field Descriptions | 54 |
| 37 | EMIF Interrupt Mask Clear Register (EIMCR) Field Descriptions | 55 |
| 38 | NAND Flash Control Register (NANDFCR) Field Descriptions | 56 |
| 39 | NAND Flash Status Register (NANDFSR) Field Descriptions | 57 |
| 40 | NAND Flash <i>n</i> ECC Register (NANDF <i>n</i> ECC) Field Descriptions..... | 58 |

Read This First

About This Manual

This document describes the asynchronous external memory interface (EMIF) in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EMIF supports a glueless interface to a variety of external devices.

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 TMS320DM357 Digital Media System-on-Chip (DMSoC). 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.

[SPRUG06](#) — ***TMS320DM357 DMSoC Video Processing Back End (VPBE) User's Guide***. Describes the video processing back end (VPBE) in the TMS320DM357 Digital Media System-on-Chip (DMSoC) video processing subsystem. Included in the VPBE is the video encoder, on-screen display, and digital LCD controller.

[SPRUG25](#) — ***TMS320DM357 DMSoC ARM Subsystem Reference Guide***. Describes the ARM subsystem in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The ARM subsystem is designed to give the ARM926EJ-S (ARM9) master control of the device. In general, the ARM is responsible for configuration and control of the device; including the video processing subsystem, and a majority of the peripherals and external memories.

[SPRUG26](#) — ***TMS320DM357 DMSoC Universal Asynchronous Receiver/Transmitter (UART) User's Guide***. This document describes the universal asynchronous receiver/transmitter (UART) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The UART peripheral performs serial-to-parallel conversion on data received from a peripheral device, and parallel-to-serial conversion on data received from the CPU.

[SPRUG27](#) — ***TMS320DM357 DMSoC Inter-Integrated Circuit (I2C) Peripheral User's Guide***. Describes the inter-integrated circuit (I2C) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The I2C peripheral provides an interface between the DMSoC and other devices compliant with the I2C-bus specification and connected by way of an I2C-bus. External components attached to this 2-wire serial bus can transmit and receive up to 8-bit wide data to and from the DMSoC through the I2C peripheral. This document assumes the reader is familiar with the I2C-bus specification.

- [SPRUG28](#)** — ***TMS320DM357 DMSoC 64-Bit Timer User's Guide***. Describes the operation of the software-programmable 64-bit timer in the TMS320DM357 Digital Media System-on-Chip (DMSoC). Timer 0 and Timer 1 are used as general-purpose (GP) timers and can be programmed in 64-bit mode, dual 32-bit unchained mode, or dual 32-bit chained mode; Timer 2 is used only as a watchdog timer. The GP timer modes can be used to generate periodic interrupts or enhanced direct memory access (EDMA) synchronization events. The watchdog timer mode is used to provide a recovery mechanism for the device in the event of a fault condition, such as a non-exiting code loop.
- [SPRUG29](#)** — ***TMS320DM357 DMSoC Serial Peripheral Interface (SPI) User's Guide***. Describes the serial peripheral interface (SPI) in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The SPI is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (1 to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communication between the DMSoC and external peripherals. Typical applications include an interface to external I/O or peripheral expansion via devices such as shift registers, display drivers, SPI EPROMs and analog-to-digital converters.
- [SPRUG30](#)** — ***TMS320DM357 DMSoC Host Port Interface (HPI) Reference Guide***. This document describes the host port interface in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The HPI provides a parallel port interface through which an external host processor can directly access the TMS320DM357 DMSoC processor's resources (configuration and program/data memories).
- [SPRUG31](#)** — ***TMS320DM357 DMSoC General-Purpose Input/Output (GPIO) User's Guide***. Describes the general-purpose input/output (GPIO) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The GPIO peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs. When configured as an input, you can detect the state of the input by reading the state of an internal register. When configured as an output, you can write to an internal register to control the state driven on the output pin.
- [SPRUG32](#)** — ***TMS320DM357 DMSoC Multimedia Card (MMC)/Secure Digital (SD) Card Controller User's Guide***. Describes the multimedia card (MMC)/secure digital (SD) card controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The MMC/SD card is used in a number of applications to provide removable data storage. The MMC/SD controller provides an interface to external MMC and SD cards. The communication between the MMC/SD controller and MMC/SD card(s) is performed by the MMC/SD protocol.
- [SPRUG33](#)** — ***TMS320DM357 DMSoC Asynchronous External Memory Interface (EMIF) User's Guide***. Describes the asynchronous external memory interface (EMIF) in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EMIF supports a glueless interface to a variety of external devices.
- [SPRUG34](#)** — ***TMS320DM357 DMSoC Enhanced Direct Memory Access (EDMA) Controller User's Guide***. Describes the operation of the enhanced direct memory access (EDMA3) controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EDMA3 controller's primary purpose is to service user-programmed data transfers between two memory-mapped slave endpoints on the DMSoC.
- [SPRUG35](#)** — ***TMS320DM357 DMSoC Audio Serial Port (ASP) User's Guide***. Describes the operation of the audio serial port (ASP) audio interface in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The primary audio modes that are supported by the ASP are the AC97 and IIS modes. In addition to the primary audio modes, the ASP supports general serial port receive and transmit operation, but is not intended to be used as a high-speed interface.
- [SPRUG36](#)** — ***TMS320DM357 DMSoC Ethernet Media Access Controller (EMAC)/Management Data Input/Output (MDIO) Module User's Guide***. Discusses the ethernet media access controller (EMAC) and physical layer (PHY) device management data input/output (MDIO) module in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The EMAC controls the flow of packet data from the DMSoC to the PHY. The MDIO module controls PHY configuration and status monitoring.

[SPRUG37](#) — TMS320DM357 DMSoC Pulse-Width Modulator (PWM) Peripheral User's Guide.

Describes the pulse-width modulator (PWM) peripheral in the TMS320DM357 Digital Media System-on-Chip (DMSoC).

[SPRUG38](#) — TMS320DM357 DMSoC DDR2 Memory Controller User's Guide. Describes the DDR2 memory controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The DDR2 memory controller is used to interface with JESD79D-2A standard compliant DDR2 SDRAM devices.

[SPRUG39](#) — TMS320DM357 DMSoC Video Processing Front End (VPFE) User's Guide. Describes the video processing front end (VPFE) in the TMS320DM357 Digital Media System-on-Chip (DMSoC) video processing subsystem. Included in the VPFE is the preview engine, CCD controller, resizer, histogram, and hardware 3A (H3A) statistic generator.

[SPRUGH2](#) — TMS320DM357 DMSoC Peripherals Overview Reference Guide. This document provides an overview of the peripherals in the TMS320DM357 Digital Media System-on-Chip (DMSoC).

[SPRUGH3](#) — TMS320DM357 DMSoC Universal Serial Bus Controller User's Guide. This document describes the universal serial bus (USB) controller in the TMS320DM357 Digital Media System-on-Chip (DMSoC). The USB controller supports data throughput rates up to 480 Mbps. It provides a mechanism for data transfer between USB devices and also supports host negotiation.

Trademarks

Asynchronous External Memory Interface (EMIF)

1 Introduction

This document describes the operation of the external memory interface (EMIF) in the TMS320DM357 Digital Media System-on-Chip (DMSoC).

1.1 Purpose of the Peripheral

The purpose of this EMIF is to provide a means to connect to a variety of external devices including:

- NAND Flash
- Asynchronous devices including Flash and SRAM
- Host processor interfaces such as the host port interface (HPI) on a Texas Instruments ARM

The most common use for the EMIF is to interface with both flash devices and SRAM devices. [Section 3](#) contains examples for operating the EMIF in this configuration.

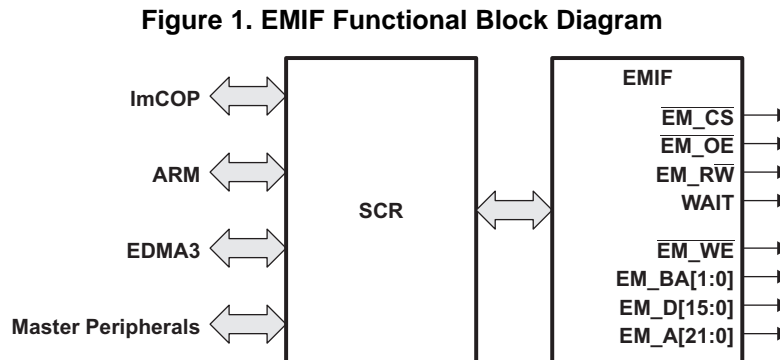
1.2 Features

The EMIF includes many features to enhance the ease and flexibility of connecting to external asynchronous devices. The EMIF features includes support for:

- 4 addressable chip select spaces of up to 32MB each
- 8-bit and 16-bit data bus widths
- Programmable cycle timings such as setup, strobe, and hold times as well as turnaround time
- Select strobe mode
- Extended Wait mode
- NAND Flash ECC generation
- Connecting as a host to a TI ARM HPI interface
- Data bus parking

1.3 Functional Block Diagram

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



2 Peripheral Architecture

This section provides details about the architecture and operation of the EMIF.

2.1 Clock Control

The EMIF's internal clock is sourced from the CLKDIV6 clock domain of PLL controller 1 and cannot be sourced directly from an external input clock. The frequency of the CLKDIV6 clock domain is the PLL1 frequency divided by 6. Changes to the frequency of the input clock to PLL controller 1 and PLL controller 1 multiplier values alters the operating frequency of the EMIF. See the *TMS320DM357 DMSoC ARM Subsystem Reference Guide* ([SPRUG25](#)) for more information on how to program the PLL controller.

2.2 EMIF Requests

Three different sources within the device can make requests to the EMIF. These requests consist of accesses to asynchronous memory and EMIF memory mapped registers. Because the EMIF can process only one request at a time, a high performance switched central resource (SCR) exists to provide prioritized requests from the different sources to the EMIF. Each requester has a programmable priority value that may be configured in the chip level MSTRPRIO register or in the 3PCC QUEPRI register. See the *TMS320DM357 DMSoC ARM Subsystem Reference Guide* ([SPRUG25](#)) for more information on these registers. The sources are:

1. EDMA3
2. ARM
3. Master Peripherals

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

2.3 Signal Descriptions

Table 1 describes the function of each of the EMIF pins.

Table 1. EMIF Pins

| Pins(s) | I/O | Description |
|---------------------------------|-----|---|
| EM_A[21:0] | O | EMIF address bus. These pins are used in conjunction with the EM_BA pins to form the address that is sent to the device. |
| EM_BA[1:0] | O | EMIF bank address. These pins are used in conjunction with the EM_A pins to form the address that is sent to the device. |
| $\overline{\text{EM_CS}}[5:2]$ | O | Active-low chip enable pin for asynchronous devices. These pins are meant to be connected to the chip-select pin of the attached asynchronous device. |
| EM_D[15:0] | I/O | EMIF data bus. |
| EM_R $\overline{\text{W}}$ | O | Read/Write select pin. This pin is high for the duration of an asynchronous read access cycle and low for the duration of an asynchronous write cycle. |
| $\overline{\text{EM_OE}}$ | 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. |
| $\overline{\text{EM_WE}}$ | O | Active-low write enable. This pin provides a signal which is active-low during the strobe period of an asynchronous write access cycle. |
| EM_WAIT | I | Wait input with programmable polarity. A connected asynchronous device can extend the strobe period of an access cycle by asserting the WAIT input to the EMIF as described in Section 2.5.8. To enable this functionality, the EW bit in the asynchronous configuration register (AnCR) must be set to 1. In addition, the WPO bit in the asynchronous wait cycle configuration register (AWCCR) must be configured to define the polarity of the EM_WAIT pin. |

2.4 Pin Multiplexing

The EMIF pins are multiplexed with other peripherals such as HPI and GPIO. Please see the device-specific data manual for instructions on how to select the EMIF pins for proper operation.

2.5 Asynchronous Controller and Interface

The EMIF easily interfaces to a variety of asynchronous devices including Flash and ASRAM. It can be operated in three major modes:

- Normal mode
- Select Strobe (SS) mode
- NAND Flash mode

The behavior of the $\overline{\text{EM_CS}}$ signal is the single difference between Normal mode and Select Strobe mode (see Table 2). In Normal mode, the $\overline{\text{EM_CS}}$ signal becomes active at the beginning of the setup period and remains active for the duration of the transfer. In Select Strobe mode, the $\overline{\text{EM_CS}}$ signal functions as a strobe signal, active only during the strobe period of an access.

In NAND Flash mode, the EMIF hardware is able to calculate the error correction code (ECC) for each 512 byte data transfer. In addition to the three modes of operation, the EMIF 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.

Table 2. Behavior of $\overline{\text{EM_CS}}$ Signal Between Normal Mode and Select Strobe Mode

| Mode | Operation of $\overline{\text{EM_CS}}[5:2]$ |
|---------------|---|
| Normal | Active during the entire asynchronous access cycle |
| Select Strobe | Active only during the strobe period of an access cycle |

2.5.1 Interfacing to Asynchronous Memory

Figure 2 shows the EMIF's external pins used in interfacing with an asynchronous device. Of special note is the connection between the EMIF and the external device's address bus. The EMIF 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 EM_BA[1] and EM_BA[0] pins provide the least-significant bits of the halfword or byte address, respectively. Figure 2 and Figure 3 show the mapping between the EMIF 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 configuration register (AnCR).

Figure 2. EMIF Asynchronous Interface

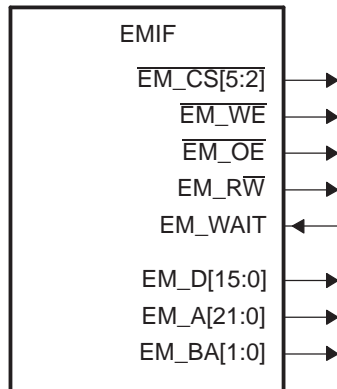
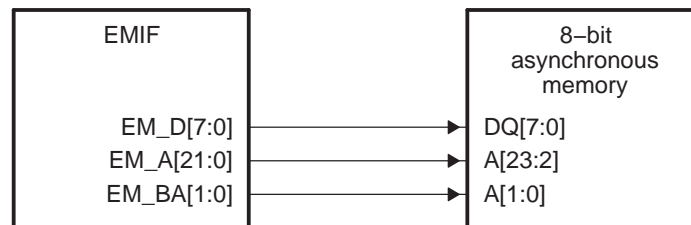
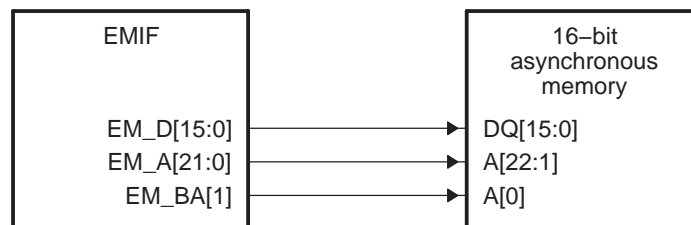


Figure 3. EMIF to 8-bit and 16-bit Memory Interfaces



a) EMIF to 8-bit memory interface



b) EMIF to 16-bit memory interface

2.5.2 Programmable Asynchronous Parameters

The EMIF allows a high degree of programmability for shaping asynchronous accesses. The programmable parameters are:

- **Setup:** The time between the beginning of a memory cycle (address valid) and the activation of the output enable or write enable strobe
- **Strobe:** The time between the activation and deactivation of output enable or write enable strobe.
- **Hold:** The time between the deactivation of output enable or write enable strobe and the end of the cycle, which may be indicated by an address change or the deactivation of the $\overline{\text{EM_CS}}$ signal.

Separate parameters are provided for read and write cycles. Each parameter is programmed in terms of EMIF clock cycles.

2.5.3 Configuring the EMIF for Asynchronous Accesses

The operation of the EMIF's asynchronous interface can be configured by programming the appropriate memory-mapped registers. The reset value and bit position for each register field can be found in [Section 4](#). The following tables list the programmable register fields and describe the purpose of each field. These registers should not be programmed while an asynchronous access is in progress. The transfer following a write to these registers will use the new configuration.

[Table 3](#) describes the asynchronous configuration register ($A_n\text{CR}$). There are four $A_n\text{CR}$ s. Each chip select space has a dedicated $A_n\text{CR}$. This allows each chip select space to be programmed independently to interface to different asynchronous memory types.

Table 3. Description of the Asynchronous Configuration Register ($A_n\text{CR}$)

| Parameter | Description |
|-------------------|---|
| SS | <p>Select Strobe mode. This bit selects the EMIF's mode of operation in the following way:</p> <ul style="list-style-type: none"> • SS = 0h selects Normal mode. $\overline{\text{EM_CS}}$ is active for duration of access. • SS = 1h selects Select Strobe mode. $\overline{\text{EM_CS}}$ acts as a strobe. |
| EW | <p>Extended Wait mode enable.</p> <ul style="list-style-type: none"> • EW = 0h disables Extended Wait mode • EW = 1h enables Extended Wait mode <p>When set to 1, the EMIF enables its Extended Wait mode in which the strobe width of an access cycle can be extended in response to the assertion of the EM_WAIT pin. The WPO bit in the asynchronous wait cycle configuration register (AWCCR) controls to polarity of EM_WAIT pin. See Section 2.5.8 for more details on this mode of operation.</p> |
| W_SETUP/R_SETUP | <p>Read/Write setup widths.</p> <p>These fields define the number of EMIF clock cycles of setup time for the address pins (EM_A and EM_BA) and asynchronous chip enable ($\overline{\text{EM_CS}}$) before the read strobe pin (READ_OE) or write strobe pin (WRITE_WE) falls, minus 1 cycle. For writes, the W_SETUP field also defines the setup time for the data pins (EM_D). Refer to the datasheet of the external asynchronous device to determine the appropriate setting for this field.</p> |
| W_STROBE/R_STROBE | <p>Read/Write strobe widths.</p> <p>These fields define the number of EMIF clock cycles between the falling and rising of the read strobe pin (READ_OE) or write strobe pin (WRITE_WE), minus 1 cycle. If Extended Wait mode is enabled by setting the EW bit in the asynchronous configuration register ($A_n\text{CR}$), 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.</p> |
| W_HOLD/R_HOLD | <p>Read/Write hold widths.</p> <p>These fields define the number of EMIF clock cycles of hold time for the address pins (EM_A and EM_BA) and asynchronous chip enable ($\overline{\text{EM_CS}}$) after the read strobe pin (READ_OE) or write strobe pin (WRITE_WE) rises, minus 1 cycle. For writes, the W_HOLD field also defines the hold time for the data pins (EM_D). Refer to the datasheet of the external asynchronous device to determine the appropriate setting for this field.</p> |
| TA | <p>Minimum turnaround time.</p> <p>This field defines the minimum number of EMIF clock cycles between the end of one asynchronous access and the start of another, minus 1 cycle. This delay is not incurred when a read is followed by a read, or a write is followed by a write to the same chip select space. The purpose of this feature is to avoid contention on the bus. Refer to the datasheet of the external asynchronous device to determine the appropriate setting for this field.</p> |

Table 3. Description of the Asynchronous Configuration Register (AnCR) (continued)

| Parameter | Description |
|-----------|--|
| ASIZE | <p>Asynchronous Device Bus Width. This field determines the data bus width of the asynchronous interface in the following way:</p> <ul style="list-style-type: none"> ASIZE = 0h selects an 8-bit bus ASIZE = 1h selects a 16-bit bus <p>The configuration of ASIZE determines the function of the EM_A and EM_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 = 0h. Refer to the datasheet of the external asynchronous device to determine the appropriate setting for this field.</p> |

Table 4. Description of the Asynchronous Wait Cycle Configuration Register (AWCCR)

| Parameter | Description |
|-----------|---|
| WPO | <p>WAIT Polarity.</p> <ul style="list-style-type: none"> WPO = 0h selects active-high polarity WPO = 1h selects active-low polarity <p>When set to 1, the EMIF will wait if the EM_WAIT pin is high. When cleared to 0, the EMIF will wait if the EM_WAIT pin is low. The EMIF must have the Extended Wait mode enabled for the EM_WAIT pin to affect the width of the strobe period.</p> |
| MEWC | <p>Maximum Extended Wait Cycles. This field configures the number of EMIF clock cycles the EMIF will wait for the EM_WAIT pin to be deactivated during the strobe period of an access cycle. The maximum number of EMIF clock cycles the EMIF will wait is determined by the following formula:</p> $\text{Maximum Extended Wait Cycles} = (\text{MEWC} + 1) \gg 16$ <p>If the EM_WAIT pin is not deactivated within the time specified by this field, the EMIF resumes the access cycle, registering whatever data is on the bus and preceding 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 EMIF interrupt mask set register (EIMSR). Refer to Section 2.5.11 for more information about the EMIF interrupts.</p> |

Table 5. Description of the EMIF Interrupt Mask Set Register (EIMSR)

| Parameter | Description |
|-----------|--|
| WRMSET | <p>Wait Rise Mask Set. Writing a 1 enables an interrupt to be generated when a rising edge on EM_WAIT occurs.</p> |
| ATMSET | <p>Asynchronous Timeout Mask Set. Writing a 1 to this bit enables an interrupt to be generated when an asynchronous timeout occurs.</p> |

Table 6. Description of the EMIF Interrupt Mast Clear Register (EIMCR)

| Parameter | Description |
|-----------|---|
| WRMCLR | <p>Wait Rise Mask Clear. Writing a 1 to this bit disables the interrupt, clearing the WRMSET bit in the EMIF interrupt mask set register (EIMSR).</p> |
| ATMCLR | <p>Asynchronous Timeout Mask Clear. Writing a 1 to this bit disables the interrupt, clearing the ATMSET bit in the EMIF interrupt mask set register (EIMSR).</p> |

2.5.4 Read and Write Operations in Normal Mode

Normal mode is the asynchronous interface's default mode of operation. The Normal mode is selected when the SS bit in the asynchronous configuration register (*AnCR*) is cleared to 0. In this mode, the *EM_CS* signal operates as a chip enable signal, active throughout the duration of the memory access.

2.5.4.1 Asynchronous Read Operations (Normal Mode)

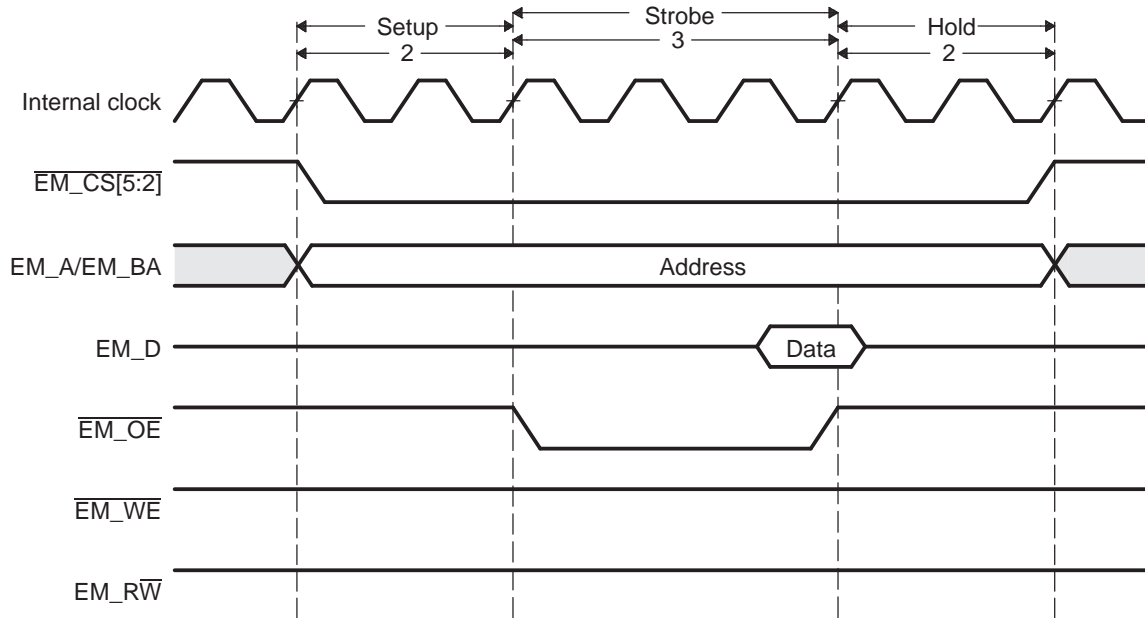
An asynchronous read is performed when any of the requesters mentioned in [Section 2.2](#) request a read from the attached asynchronous memory. In the event that the read request cannot be serviced by a single access cycle to the external device, multiple access cycles will be performed by the EMIF until the entire request is fulfilled. The details of an asynchronous read operation in Normal mode are described in [Table 7](#) and an example timing diagram of a basic read operation is shown in [Figure 4](#).

Note: During the entirety of an asynchronous read operation, the $\overline{\text{WRITE_WE}}$ and $\overline{\text{EM_RW}}$ pins are driven high.

Table 7. Asynchronous Read Operation in Normal Mode

| Time Interval | Pin Activity in WE Strobe Mode |
|------------------------|---|
| Turnaround period | <p>Once the EMIF receives a read request, the EMIF 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 configuration register (<i>AnCR</i>). There are two exceptions to this rule:</p> <ul style="list-style-type: none"> • If the current read operation was directly preceded by another read operation to the same CS space, no turnaround cycles are inserted. • If the current read operation was not directly preceded by a read operation to the same CS space and the TA field has been cleared to 0, one turn-around cycle will be inserted. <p>After the EMIF has waited for the turnaround cycles to complete, it proceeds to the setup period of the operation.</p> |
| Start of setup period | <p>At the beginning of the setup period:</p> <ul style="list-style-type: none"> • The setup, strobe, and hold values are set according to the R_SETUP, R_STROBE, and R_HOLD values in <i>AnCR</i>. • The address pins EM_A and EM_BA become valid • $\overline{\text{EM_CS}}$ falls to enable the external device (if not already low from a previous operation) |
| Start of strobe period | <p>At the beginning of the strobe period</p> <ul style="list-style-type: none"> • $\overline{\text{READ_OE}}$ falls |
| Start of hold period | <p>At the beginning of the hold period:</p> <ul style="list-style-type: none"> • $\overline{\text{READ_OE}}$ rises • The EMIF samples the data on the EM_D bus. |
| End of hold period | <p>At the end of the hold period:</p> <ul style="list-style-type: none"> • The address pins EM_A and EM_BA become invalid • $\overline{\text{EM_CS}}$ rises (if no more operations are required to complete the current request) <p>The EMIF will 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 EMIF 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 EMIF 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 EMIF instead enters directly into the turnaround period for the pending read or write operation.</p> |

Figure 4. Timing Waveform of an Asynchronous Read Cycle in Normal Mode



2.5.4.2 Asynchronous Write Operations (Normal Mode)

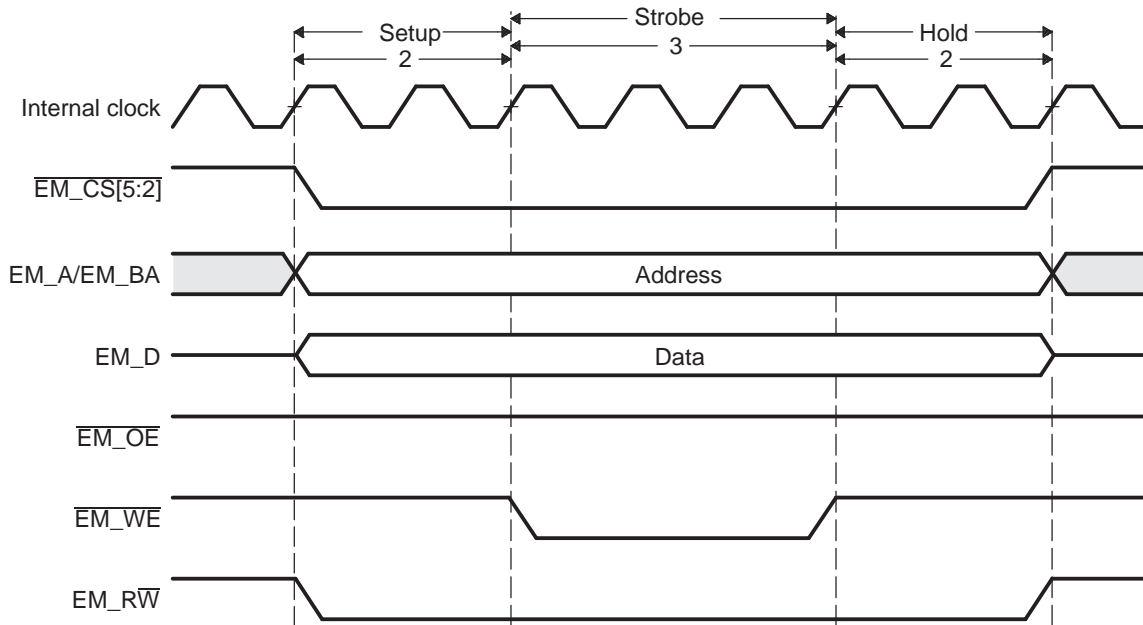
An asynchronous write is performed when any of the requesters mentioned in [Section 2.2](#) request a write to asynchronous memory. In the event that the write request cannot be serviced by a single access cycle to the external device, multiple access cycles will be performed by the EMIF until the entire request is fulfilled. The details of an asynchronous write operation in Normal mode are described in [Table 8](#) and an example timing diagram of a basic write operation is shown in [Figure 5](#).

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

Table 8. Asynchronous Write Operation in Normal Mode

| Time Interval | Pin Activity in WE Strobe Mode |
|------------------------|---|
| Turnaround period | <p>Once the EMIF receives a write request, the EMIF 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 configuration register (AnCR). There are two exceptions to this rule:</p> <ul style="list-style-type: none"> • If the current write operation was directly preceded by another write operation to the same CS space, no turnaround cycles are inserted. • If the current write operation was not directly preceded by a write operation to the same CS space and the TA field has been cleared to 0, one turnaround cycle will be inserted. <p>After the EMIF has waited for the turnaround cycles to complete, it proceeds to the setup period of the operation.</p> |
| Start of setup period | <p>At the beginning of the setup period:</p> <ul style="list-style-type: none"> • The setup, strobe, and hold values are set according to the W_SETUP, W_STROBE, and W_HOLD values in AnCR. • The address pins EM_A and EM_BA and the data pins EM_D become valid. • The EM_R$\overline{\text{W}}$ pin falls to indicate a write (if not already low from a previous operation). • $\overline{\text{EM_CS}}$ falls to enable the external device (if not already low from a previous operation). |
| Start of strobe period | <p>At the beginning of the strobe period of a write operation:</p> <ul style="list-style-type: none"> • $\overline{\text{EM_WE}}$ falls |
| Start of hold period | <p>At the beginning of the hold period</p> <ul style="list-style-type: none"> • $\overline{\text{EM_WE}}$ rises |
| End of hold period | <p>At the end of the hold period:</p> <ul style="list-style-type: none"> • The address pins EM_A and EM_BA become invalid • The data pins become invalid • The EM_R$\overline{\text{W}}$ pin rises (if no more operations are required to complete the current request) • $\overline{\text{EM_CS}}$ rises (if no more operations are required to complete the current request) <p>The EMIF 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 EMIF 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 EMIF returns to its previous state unless another asynchronous request has been submitted. If this is the case, the EMIF instead enters directly into the turnaround period for the pending read or write operation.</p> |

Figure 5. Timing Waveform of an Asynchronous Write Cycle in Normal Mode



2.5.5 Read and Write Operations in Select Strobe Mode

Select Strobe mode is the EMIF's second mode of operation. The SS mode is selected when the SS bit in the asynchronous configuration register (*AnCR*) is set to 1. In this mode, the EM_CS pin functions as a strobe signal and is therefore only active during the strobe period of an access cycle.

2.5.5.1 Asynchronous Read Operations (Select Strobe Mode)

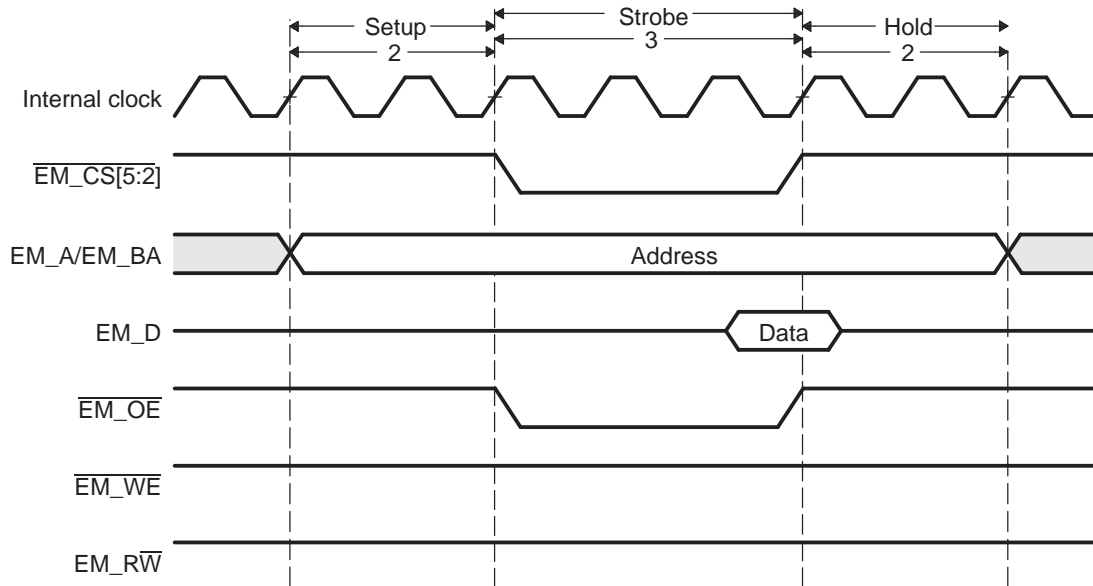
An asynchronous read is performed when any of the requesters mentioned in [Section 2.2](#) request a read from the attached asynchronous memory. In the event that the read request cannot be serviced by a single access cycle to the external device, multiple access cycles will be performed by the EMIF until the entire request is fulfilled. The details of an asynchronous read operation in Select Strobe mode are described in [Table 9](#) and an example timing diagram of a basic read operation is shown in [Figure 6](#).

Note: During the entirety of an asynchronous read operation, the $\overline{\text{EM_WE}}$ and EM_R $\overline{\text{W}}$ pins are driven high.

Table 9. Asynchronous Read Operation in Select Strobe Mode

| Time Interval | Pin Activity in Select Strobe Mode |
|------------------------|--|
| Turnaround period | <p>Once the EMIF receives a read request, the EMIF 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 configuration register (<i>AnCR</i>). There are two exceptions to this rule:</p> <ul style="list-style-type: none"> • If the current read operation was directly preceded by another read operation to the same CS space, no turnaround cycles are inserted. • If the current read operation was not directly preceded by a read operation to the same CS space and the TA field has been cleared to 0, one turnaround cycle will be inserted. <p>After the EMIF has waited for the turnaround cycles to complete, it proceeds to the setup period of the operation.</p> |
| Start of setup period | <p>At the beginning of the setup period:</p> <ul style="list-style-type: none"> • The setup, strobe, and hold values are set according to the R_SETUP, R_STROBE, and R_HOLD values in <i>AnCR</i>. • The address pins EM_A and EM_BA become valid. |
| Start of strobe period | <p>At the beginning of the strobe period:</p> <ul style="list-style-type: none"> • $\overline{\text{EM_CS}}$ and $\overline{\text{EM_OE}}$ fall at the start of the strobe period |
| Start of hold period | <p>At the beginning of the hold period:</p> <ul style="list-style-type: none"> • $\overline{\text{EM_CS}}$ and $\overline{\text{EM_OE}}$ rise • The EMIF samples the data on the EM_D bus |
| End of hold period | <p>At the end of the hold period:</p> <ul style="list-style-type: none"> • The address pins EM_A and EM_BA become invalid <p>The EMIF 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 EMIF 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 EMIF returns to its previous state unless another asynchronous request has been submitted. If this is the case, the EMIF instead enters directly into the turnaround period for the pending read or write operation.</p> |

Figure 6. Timing Waveform of an Asynchronous Read Cycle in Select Strobe Mode



2.5.5.2 Asynchronous Write Operations (Select Strobe Mode)

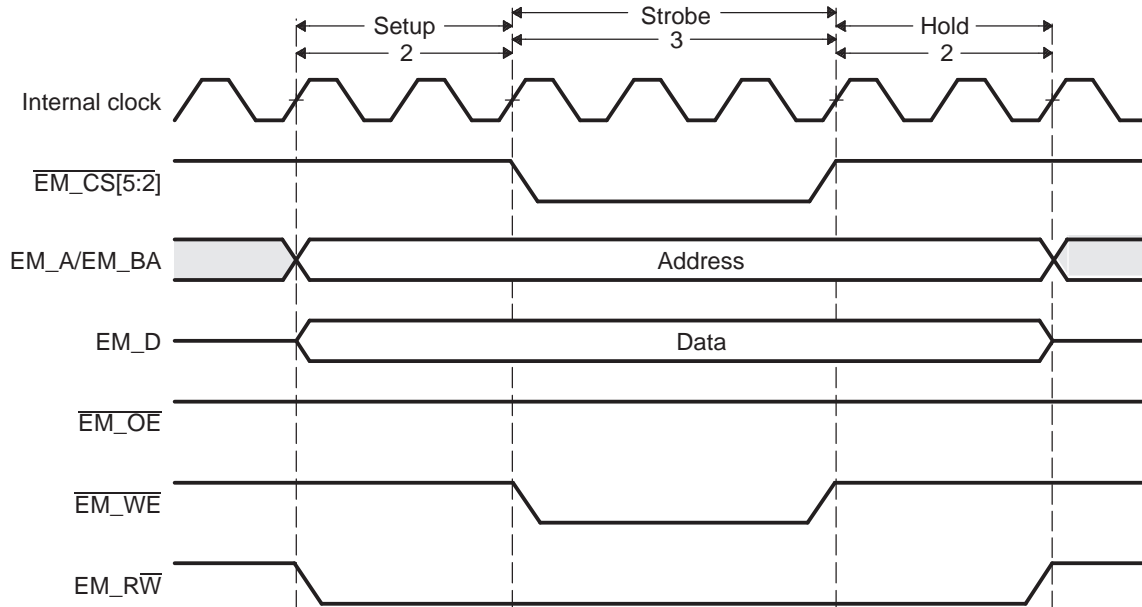
An asynchronous write is performed when any of the requesters mentioned in [Section 2.2](#) request a write to memory in the asynchronous bank of the EMIF. In the event that the write request cannot be serviced by a single access cycle to the external device, multiple access cycles will be performed by the EMIF until the entire request is fulfilled. The details of an asynchronous write operation in Select Strobe mode are described in [Table 10](#) and an example timing diagram of a basic write operation is shown in [Figure 7](#).

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

Table 10. Asynchronous Write Operation in Select Strobe Mode

| Time Interval | Pin Activity in Select Strobe Mode |
|------------------------|--|
| Turnaround period | <p>Once the EMIF receives a write request, the EMIF 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 configuration register (AnCR). There are two exceptions to this rule:</p> <ul style="list-style-type: none"> • If the current write operation was directly preceded by another write operation to the same CS space, no turnaround cycles are inserted. • If the current write operation was directly preceded by a write operation to the same CS space and the TA field has been cleared to 0, one turnaround cycle will be inserted. <p>After the EMIF has waited for the turnaround cycles to complete, it proceeds to the setup period of the operation.</p> |
| Start of setup period | <p>At the beginning of the setup period:</p> <ul style="list-style-type: none"> • The setup, strobe, and hold values are set according to the W_SETUP, W_STROBE, and W_HOLD values in AnCR. • The address pins EM_A and EM_BA and the data pins EM_D become valid. • The EM_R$\overline{\text{W}}$ pin falls to indicate a write (if not already low from a previous operation). |
| Start of strobe period | <p>At the beginning of the strobe period:</p> <ul style="list-style-type: none"> • $\overline{\text{EM_CS}}$ and $\overline{\text{EM_WE}}$ fall |
| Start of hold period | <p>At the beginning of the hold period:</p> <ul style="list-style-type: none"> • $\overline{\text{EM_CS}}$ and $\overline{\text{EM_WE}}$ rise |
| End of hold period | <p>At the end of the hold period:</p> <ul style="list-style-type: none"> • The address pins EM_A and EM_BA become invalid • The data pins become invalid • The EM_R$\overline{\text{W}}$ pin rises (if no more operations are required to complete the current request) <p>The EMIF 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 EMIF 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 EMIF returns to its previous state unless another asynchronous request has been submitted. If this is the case, the EMIF instead enters directly into the turn-around period for the pending read or write operation.</p> |

Figure 7. Timing Waveform of an Asynchronous Write Cycle in Select Strobe Mode



2.5.6 NAND Flash Mode

NAND Flash mode is the EMIF's third mode of operation. Each chip select space may be placed in NAND Flash mode individually by setting the appropriate CS_nNAND bit in the NAND Flash control register (NANDFCR). [Table 11](#) 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 EMIF hardware can calculate the error correction code (ECC) for each 512 byte data transfer to that chip select space. The EMIF 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 EMIF can execute a complete NAND access cycle.

Table 11. Description of the NAND Flash Control Register (NANDFCR)

| Parameter | Description |
|-----------|--|
| CS5ECC | NAND Flash ECC state for chip select 5. <ul style="list-style-type: none"> Set to 1 to start an ECC calculation. Cleared to 0 when NAND Flash 4 ECC register (NANDF1ECC) is read. |
| CS4ECC | NAND Flash ECC state for chip select 4. <ul style="list-style-type: none"> Set to 1 to start an ECC calculation. Cleared to 0 when NAND Flash 3 ECC register (NANDF1ECC) is read. |
| CS3ECC | NAND Flash ECC state for chip select 3. <ul style="list-style-type: none"> Set to 1 to start an ECC calculation. Cleared to 0 when NAND Flash 2 ECC register (NANDF1ECC) is read. |
| CS2ECC | NAND Flash ECC state for chip select 2. <ul style="list-style-type: none"> Set to 1 to start an ECC calculation. Cleared to 0 when NAND Flash 1 ECC register (NANDF1ECC) is read. |
| CS5NAND | NAND Flash mode for chip select 5. <ul style="list-style-type: none"> Set to 1 to enable NAND Flash mode. |
| CS4NAND | NAND Flash mode for chip select 4. <ul style="list-style-type: none"> Set to 1 to enable NAND Flash mode. |
| CS3NAND | NAND Flash mode for chip select 3. <ul style="list-style-type: none"> Set to 1 to enable NAND Flash mode. |
| CS2NAND | NAND Flash mode for chip select 2. <ul style="list-style-type: none"> Set to 1 to enable NAND Flash mode. |

2.5.6.1 Configuring for NAND Flash Mode

Similar to the asynchronous accesses previously described, the EMIF's memory-mapped registers must be programmed appropriately to interface to a NAND Flash device. [Table 12](#) lists the bit fields that must be programmed when operating in NAND Flash mode and the values to set each bit. NAND Flash mode cannot be used with Extended Wait mode.

Table 12. Configuration For NAND Flash

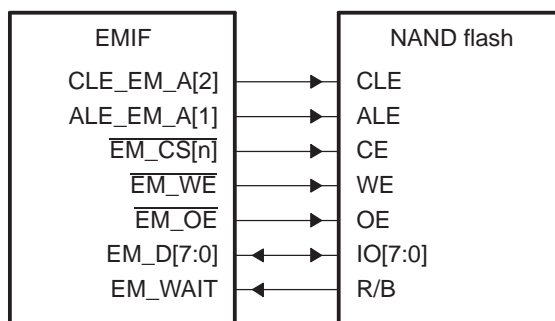
| Register | Bit Field | Configuration Value |
|--|-------------------|--|
| Asynchronous configuration register (AnCR) | SS | 0 |
| | EW | 0 |
| | W_SETUP/R_SETUP | See Section 3.2 for information on how to program. |
| | W_STROBE/R_STROBE | See Section 3.2 for information on how to program. |
| | W_HOLD/R_HOLD | See Section 3.2 for information on how to program. |
| | ASIZE | Programmed to equal the width of the NAND Flash device |
| NAND Flash control register (NANDFCR) | CS2NAND | 1 |

2.5.6.2 Connecting to NAND Flash

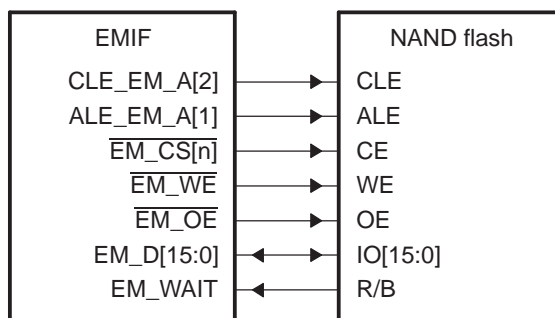
Figure 8 shows the EMIF external pins used to interface with a NAND Flash device. EMIF address lines are used to drive the NAND Flash device's command latch enable (CLE) and address latch enable (ALE) signals. Any EMIF address line may be used to drive the CLE and ALE signals of the NAND Flash. However, it is recommended, especially when booting from NAND Flash, that EM_A[2:1] be used. This is because these pins are not muxed with another peripheral and are therefore always available.

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

Figure 8. EMIF 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 EMIF always drives the least significant bit of a 32-bit word address on EM_A[0]. This functionality must be considered when attempting to drive the address lines connected to CLE and ALE to the appropriate state.

For example, if using EM_A[2] and EM_A[1] to connect to CLE and ALE, respectively, the following offsets should be chosen:

- 00h to drive CLE and ALE low
- 10h to drive CLE high and ALE low
- 0Bh to drive CLE low and ALE high

These offsets should be added to the base address for the chip select space the NAND Flash device is connected to. For example, if the base address of the CS space the NAND Flash device is connected to is 0200 0000h, then the above list translates to the following memory-mapped addresses: 0200 0000h, 0200 0010h, and 0200 000Bh, respectively. Therefore, when attempting to drive CLE high and ALE low, the memory-mapped address of 0200 0010h would be written to.

2.5.6.4 NAND Read and Program Operations

A NAND Flash access cycle is composed of a command, address, and data phase. The EMIF 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 (as described above) must be completed by the EMIF. 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 EMIF 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. See [Section 2.5.6.5](#) for more details.

Note: The EMIF 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 DMA

When performing NAND accesses, the EDMA 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's ability to transfer larger quantities of data with a single request. In this section we will focus on using the EDMA 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. These are:

- CLE_EM_A[2] and ALE_EM_A[1] are lower address lines and must be driven low
- The EMIF does not support a constant address mode, but only supports linear incrementing address modes.

Since the EMIF 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, since the CLE and ALE are driven by lower address lines, 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. The proper EDMA configurations are described below.

EDMA setup for a NAND Flash data read:

- ACNT \leq 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:

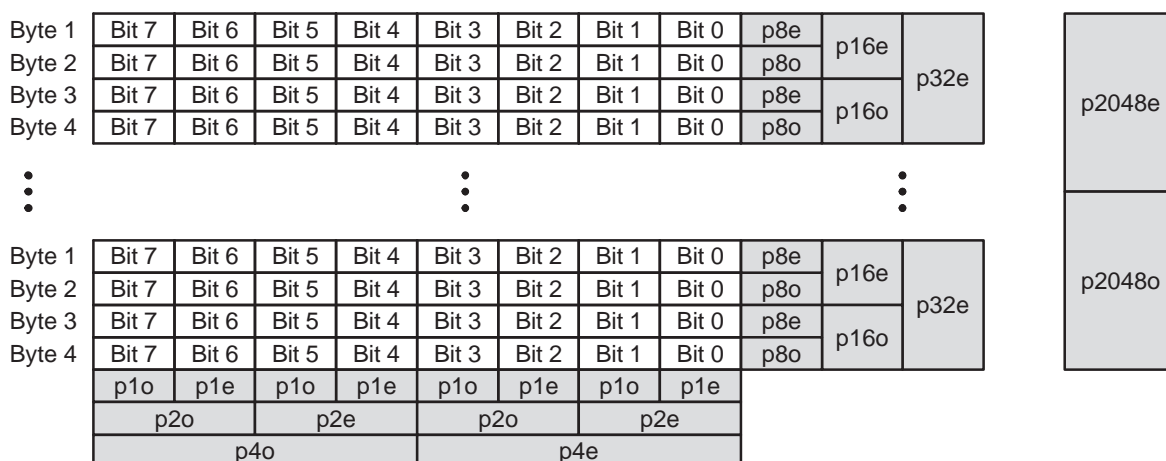
- ACNT \leq 8 bytes (this can also be set to less than or equal to the external data bus width)
- BCNT = transfer size in bytes/ACNT
- SIDX (source index) = ACNT
- DIDX (destination index) = 0
- AB synchronized

2.5.6.6 ECC Generation

If the CS n NAND bit in the NAND Flash control register (NANDFCR) is set to 1, the EMIF supports ECC calculation for up to 512 bytes for the corresponding chip select care. To perform the ECC calculation, the CS2ECC bit in NANDFCR must be set to 1. The ECC calculation for each chip select space is independent of each other. It is the responsibility of the software to start the ECC calculation by writing to the CS2ECC 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 1 ECC register (NANDF1ECC) once the transfer to NAND Flash has completed. If the software writes or reads more than 512 bytes, the ECC will be incorrect. There is a NANDF n ECC for each chip select space and when read, the corresponding CS n ECC bit in NANDFCR is cleared. The NANDF1ECC is cleared upon writing a 1 to the CS2ECC bit. [Figure 9](#) shows the algorithm used to calculate the ECC value for an 8-bit NAND Flash.

For an 8-bit NAND Flash p1e through p4e are column parities and p8e through p2048 are row parities. Similarly, the algorithm can be extended to a 16-bit NAND Flash. For a 16-bit NAND Flash p1e through p8e are column parities and p16e through p2048 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 9. ECC Value for 8-Bit NAND Flash



2.5.6.7 NAND Flash Status Register (NANDFSR)

The NAND Flash status register (NANDFSR) indicates the raw status of the EM_WAIT pin. The EM_WAIT pin should be connected to the NAND Flash device's R/B signal, so that it indicates whether or not the NAND Flash device is busy. During a read, the R/B signal will transition and remain low while the NAND Flash retrieves the data requested. Once the R/B signal transitions high, the requested data is ready and should be read by the EMIF. During a write/program operation, the R/B signal transitions and remains low while the NAND Flash is programming the Flash with the data it has received from the EMIF. Once the R/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 EM_WAIT pin, the EMIF sets the WR (wait rise) bit in the EMIF interrupt raw register (EIRR). Therefore, the EMIF wait rise interrupt may be used to indicate the status of the NAND Flash device. The WP0 bit in the asynchronous wait cycle configuration register (AWCCR) does not affect the NAND Flash status register (NANDFSR) or the WR bit in EIRR. See [Section 2.5.11.1](#) for more a detailed description of the wait rise interrupt.

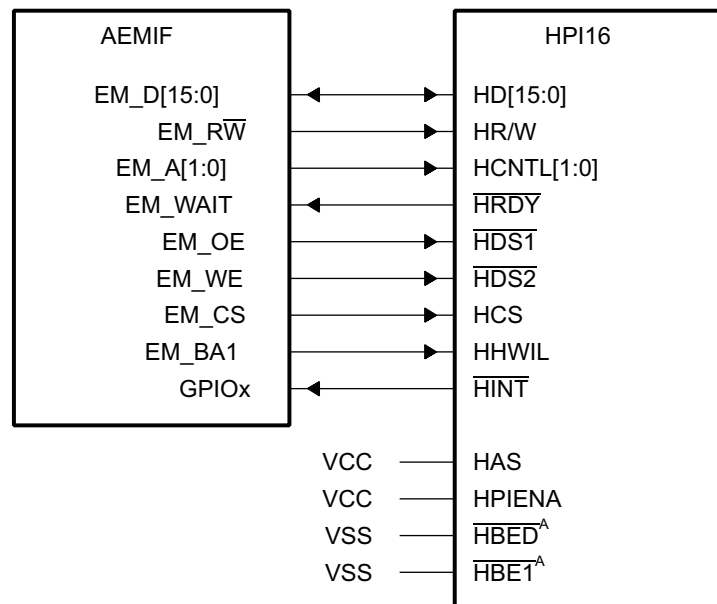
2.5.6.8 Interfacing to a Non-CE Don't Care NAND Flash

As explained in [Section 2.5.6.4](#), the EMIF 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 will configure the selected GPIO to be high. See [Section 3](#) for more details on the GPIO workaround.

2.5.7 Interfacing to a TI HPI

The EMIF supports connecting as a host to a TI HPI interface. When connecting to a TI HPI interface, the EMIF must be configured for normal mode operation. [Figure 10](#) shows the connection diagram.

Figure 10. EMIF to 16-Bit Multiplexed HPI16 Interface



A HBE signals may not be present on all HPI interfaces.

2.5.8 Extended Wait Mode and the EM_WAIT Pin

The Extended Wait mode 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 configuration register (AnCR). When the EW bit is set, the EMIF monitors the EM_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 EMIF detects that the EM_WAIT pin has been asserted, it will begin inserting extra strobe cycles into the operation until the EM_WAIT pin is deactivated by the external device. The EMIF will then return to the last cycle of the programmed strobe period and the operation will proceed as usual from this point. Refer to the device-specific data manual for details on the timing requirements of the EM_WAIT signal.

The EM_WAIT pin cannot be used to extend the strobe period indefinitely. The programmable MEWC bit in the asynchronous wait cycle configuration register (AWCCR) determines the maximum number of EMIF clock cycles the strobe period may be extended beyond the programmed length. When the number of cycles programmed in the MEWC bit expires, the EMIF proceeds to the hold period of the operation regardless of the state of the EM_WAIT pin. The EMIF can also generate an interrupt upon expiration of this counter. See [Section 2.5.11.1](#) for details on enabling this interrupt.

For the EMIF to function properly in the Extended Wait mode, the WP0 bit in AWCCR must be programmed to match the polarity of the attached device. When WP0 is in its reset state of 1, the EMIF will insert wait cycles when the EM_WAIT pin is sampled high; when WP0 is cleared to 0, the EMIF will insert wait cycles only when the EM_WAIT pin is sampled low. This programmability allows for a glueless connection to larger variety of asynchronous devices.

Finally, a restriction is placed on the setup and strobe period timing parameters when operating in Extended Wait mode. Specifically, the sum of the W_SETUP and W_STROBE fields must be greater than 4, and the sum of the R_SETUP and R_STROBE fields must be greater than 4 for the EMIF to recognize the EM_WAIT pin has been asserted. The W_SETUP, W_STROBE, R_SETUP, and R_STROBE fields are in AnCR.

2.5.9 Data Bus Parking

The EMIF 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 EMIF issues a read command to the external memory does it stop driving the data bus. After the EMIF latches the last read data, it immediately parks the data bus again.

2.5.10 Reset and Initialization Considerations

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

1. The $\overline{\text{RESET}}$ pin on the device is asserted
2. The EMIF is placed in reset by the Power and Sleep Controller.

When a reset occurs, the EMIF will immediately abandon any access request that is in progress and reset all registers and internal logic to their default state.

Following device power up and deassertion of the $\overline{\text{RESET}}$ pin, the internal clock to the EMIF is turned on and the EMIF memory-mapped registers are programmed to their default values.

2.5.11 Interrupt Support

The EMIF supports a single interrupt to the CPU. [Section 2.5.11.1](#) details the generation and internal masking of EMIF interrupts and [Section 2.5.11.2](#) describes how the EMIF interrupts are sent to the CPU.

2.5.11.1 Interrupt Events

There are two conditions that may cause the EMIF to generate an interrupt to the CPU. These two conditions are:

- A rising edge on the EM_WAIT signal (wait rise interrupt)
- An asynchronous time out

The wait rise interrupt is not affected by the WP0 bit in the asynchronous wait cycle configuration register (AWCCR). The asynchronous time out interrupt condition occurs when the attached asynchronous device fails to deassert the EM_WAIT pin within the number of cycles defined by the MEWC bit in AWCCR.

Only when the interrupt is enabled by setting the appropriate bit (WRMSET or ATMSET) in the EMIF interrupt mask set register (EIMSR) 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 EMIF interrupt mask clear register (EIMCR). The bit fields in both the EIMSR and EIMCR may be used to indicate whether the interrupt is enabled. When the interrupt is enabled, the corresponding bit field in both the EIMSR and EIMCR will have a value of 1; when the interrupt is disabled, the corresponding bit field will have a value of 0.

The EMIF interrupt raw register (EIRR) and the EMIF interrupt mask register (EIMR) indicate the status of each interrupt. The appropriate bit (WR or AT) in EIRR is set when the interrupt condition occurs, whether or not the interrupt has been enabled. Whereas, the appropriate bit (WRM or ATM) in EIMR is set only when the interrupt condition occurs and the interrupt is enabled. Writing a 1 to the bit in EIRR clears the EIRR bit as well as the corresponding bit in EIMR.

[Table 13](#) contains a brief summary of the interrupt status and control bit fields. See [Section 4](#) for complete details on the register fields.

Table 13. Interrupt Monitor and Control Bit Fields

| Register Name | Bit Name | Description |
|--|----------|--|
| EMIF interrupt raw register (EIRR) | WR | This bit is always set when a rising edge on the EM_WAIT signal occurs. Writing a 1 clears the WR bit as well as the WRM bit in EIMR. |
| | AT | This bit is always set when an asynchronous timeout occurs. Writing a 1 clears the AT bit as well as the ATM bit in EIMR. |
| EMIF interrupt mask register (EIMR) | WRM | This bit is only set when a rising edge on the EM_WAIT signal occurs and the interrupt has been enabled by writing a 1 to the WRMSET bit in EIMSR. |
| | ATM | This bit is only set when an asynchronous timeout occurs and the interrupt has been enabled by writing a 1 to the ATMSET bit in EIMSR. |
| EMIF interrupt mask set register (EIMSR) | WRMSET | Writing a 1 to this bit enables the wait rise interrupt. |
| | ATMSET | Writing a 1 to this bit enables the asynchronous timeout interrupt. |
| EMIF interrupt mask clear register (EIMCR) | WRMCLR | Writing a 1 to this bit disables the wait rise interrupt. |
| | ATMCLR | Writing a 1 to this bit disables the asynchronous timeout interrupt. |

2.5.11.2 Interrupt Multiplexing

The EMIF interrupt is supported by the ARM. The interrupt is not multiplexed with another interrupt and is therefore always available.

2.5.12 Program Execution

Since the EMIF does not have byte enable or data mask pins, byte accesses to memory are not supported when the data bus width is equal to 16 bits. When performing data accesses on a 16-bit bus, this may be worked around by performing a write modify read back operation. When executing code from the EMIF, the bus width must be configured to be an 8-bit data bus.

2.5.13 Power Management

Power dissipation to the EMIF may be managed by gating the input clock to the EMIF off. The input clock is turned off outside of the EMIF through the use of the Power and Sleep Controller (PSC). When the PSC sends a clock stop request to the EMIF, the EMIF will complete pending transfers before issuing a clock stop acknowledge, allowing the PSC to stop the clock. See the *TMS320DM357 DMSoC ARM Subsystem Reference Guide* ([SPRUG25](#)) for more information.

2.5.14 Emulation Considerations

The operation of the EMIF is not affected when a breakpoint is reached or an emulation halt occurs.

3 Example Configuration

The EMIF allows a high degree of programmability for shaping asynchronous accesses. As previously stated, the shape and duration of the asynchronous access is determined by controlling the widths of the SETUP, STROBE, HOLD, and turnaround periods. The widths of these periods are configured by programming the asynchronous configuration register (A_nCR) for the corresponding chip select space. See [Section 2.5.3](#) and [Section 4.2](#) for more information.

The programmability inherent to the EMIF, provides the EMIF with the flexibility to interface with a variety of asynchronous memory types. By programming the W_SETUP/R_SETUP , W_STROBE/R_STROBE , W_HOLD/R_HOLD , TA, and ASIZE fields in A_nCR , the EMIF can be configured to meet the data sheet specification for most asynchronous memory devices.

This section presents examples describing how to interface the EMIF to asynchronous SRAM and NAND Flash devices.

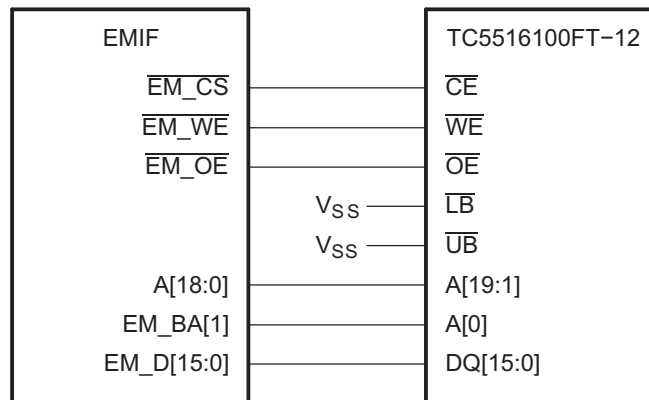
3.1 Interfacing to Asynchronous SRAM (ASRAM)

The following example describes how to interface the EMIF to the Toshiba TC55V16100FT-12 device.

3.1.1 Connecting to ASRAM

[Figure 11](#) shows how to connect the EMIF to the TC55V16100FT-12 device. Since the EMIF does not include data mask or byte enable signals, the \overline{LB} and \overline{UB} signals of the ASRAM must be tied high.

Figure 11. Connecting the EMIF to the TC55V16100FT-12



3.1.2 Meeting AC Timing Requirements for ASRAM

When configuring the EMIF to interface to ASRAM, you must consider the AC timing requirements of the ASRAM as well as the AC timing requirements of the EMIF. These can be found in the data sheet for each respective device. The read and write asynchronous cycles are programmed separately in the asynchronous configuration register (AnCR).

For a read access, [Table 14](#) to [Table 16](#) list the AC timing specifications that must be considered.

Table 14. EMIF Input Timing Requirements

| Parameter | Description |
|-----------------|--|
| t _{SU} | Data Setup time, data valid before $\overline{\text{EM_OE}}$ high |
| t _H | Data Hold time, data valid after $\overline{\text{EM_OE}}$ high |

Table 15. ASRAM Output Timing Characteristics

| Parameter | Description |
|------------------|--|
| t _{ACC} | Address Access time |
| t _{OH} | Output data Hold time for address change |
| t _{COD} | Output Disable time from chip enable |

Table 16. ASRAM Input Timing Requirement for a Read

| Parameter | Description |
|-----------------|-----------------|
| t _{RC} | Read Cycle time |

[Figure 12](#) shows an asynchronous read access and describes how the EMIF and ASRAM AC timing requirements work together to define the values for R_SETUP, R_STROBE, and R_HOLD.

From [Figure 12](#), the following equations may be derived. t_{cyc} is the period at which the EMIF operates. The R_SETUP, R_STROBE, and R_HOLD fields are programmed in terms of EMIF cycles where as the data sheet specifications are typically given in nano seconds. This explains the presence of t_{cyc} in the denominator of the following equations. A minus 1 is included in the equations because each field in AnCR is programmed in terms of EMIF clock cycles, minus 1 cycle. For example, R_SETUP is equal to R_SETUP width in EMIF clock cycles minus 1 cycle.

$$\text{R_SETUP} + \text{R_STROBE} \geq \frac{(t_{\text{ACC}}(m) + t_{\text{SU}})}{t_{\text{cyc}}} - 1$$

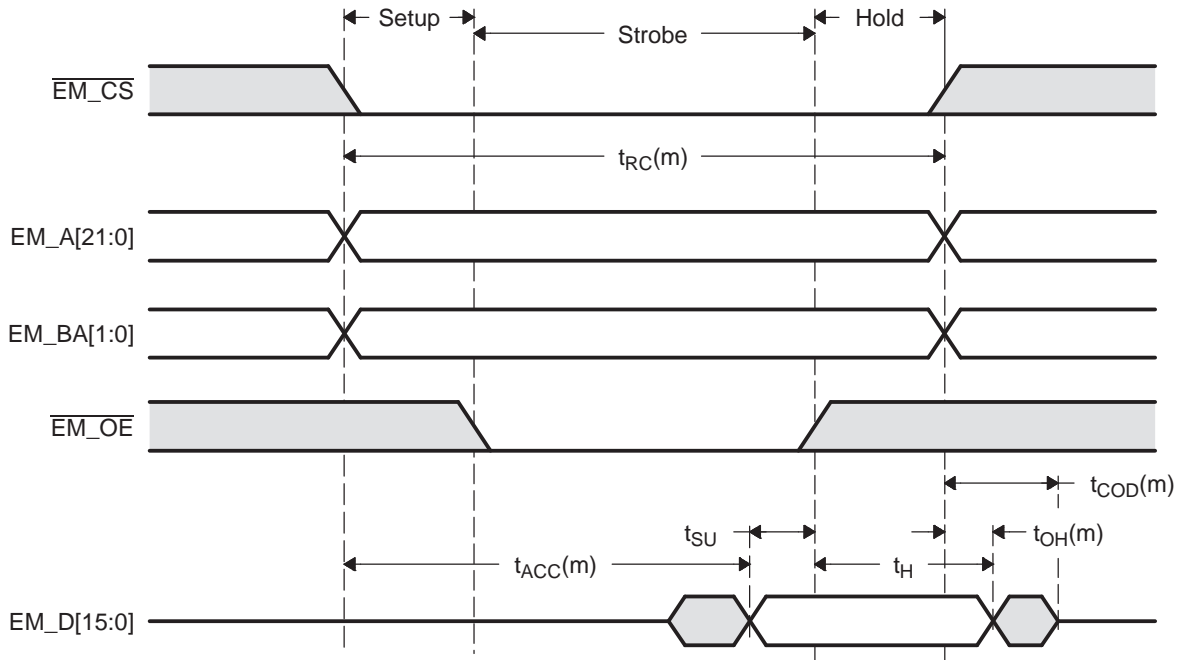
$$\text{R_SETUP} + \text{R_STROBE} + \text{R_HOLD} \geq \frac{t_{\text{RC}}(m)}{t_{\text{cyc}}} - 3$$

$$\text{R_HOLD} \geq \frac{(t_{\text{H}} - t_{\text{OH}}(m))}{t_{\text{cyc}}} - 1$$

The EMIF offers an additional parameter, TA, that defines the turnaround time between read and write cycles. This parameter protects against the situation when the output turn-off time of the memory is longer than the time it takes to start the next write cycle. If this is the case, the EMIF will drive data at the same time as the memory, causing contention on the bus. By examining [Figure 12](#), the equation for TA can be derived as:

$$\text{TA} \geq \frac{t_{\text{COD}}(m)}{t_{\text{cyc}}} - 1$$

Figure 12. Timing Waveform of an ASRAM Read



For a write access, [Table 17](#) lists the AC timing specifications that must be satisfied.

Table 17. ASRAM Input Timing Requirements for a Write

| Parameter | Description |
|-----------|-------------------------------|
| t_{WP} | Write Pulse width |
| t_{AW} | Address valid to end of Write |
| t_{DS} | Data Setup time |
| t_{WR} | Write Recovery time |
| t_{DH} | Data Hold time |
| t_{WC} | Write Cycle time |

Figure 13 shows an asynchronous write access and describes how the EMIF and ASRAM AC timing requirements work together to define values for W_SETUP, W_STROBE, and W_HOLD.

From Figure 13, the following equations may be derived. t_{cyc} is the period at which the EMIF operates. The W_SETUP, W_STROBE, and W_HOLD fields are programmed in terms of EMIF cycles where as the data sheet specifications are typically given in nano seconds. This explains the presence of t_{cyc} in the denominator of the following equations. A minus 1 is included in the equations because each field in AnCR is programmed in terms of EMIF clock cycles, minus 1 cycle. For example, W_SETUP is equal to W_SETUP width in EMIF clock cycles minus 1 cycle.

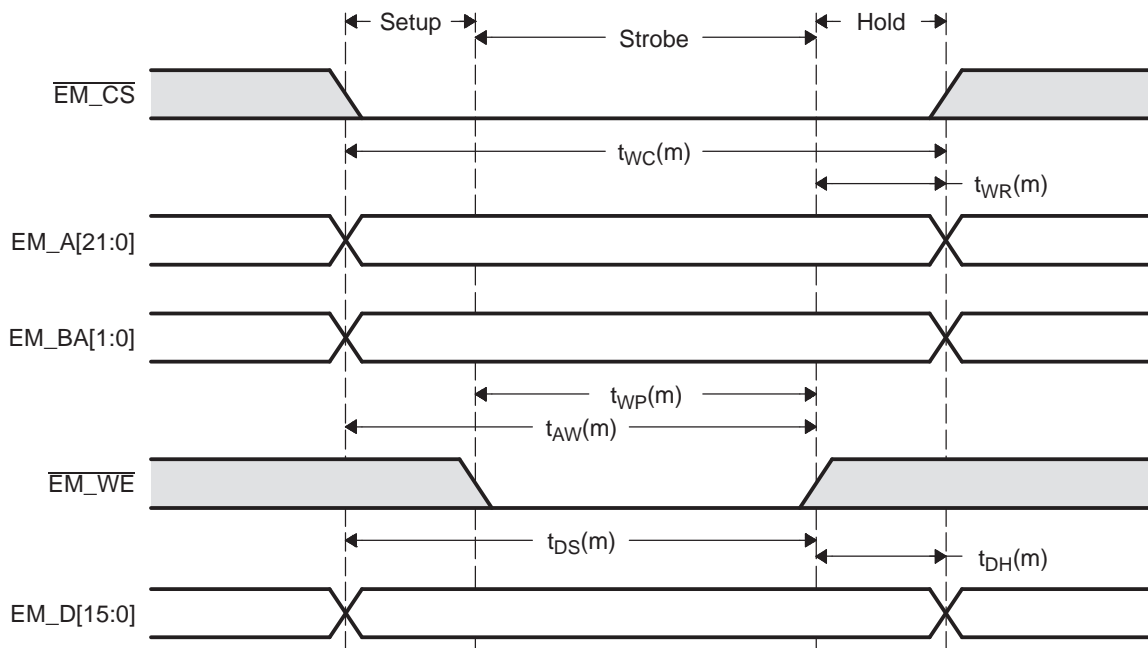
$$W_STROBE \geq \frac{t_{WP}(m)}{t_{cyc}} - 1$$

$$W_SETUP + W_STROBE \geq \max\left(\frac{t_{AW}(m)}{t_{cyc}}, \frac{t_{DS}(m)}{t_{cyc}}\right) - 1$$

$$W_HOLD \geq \max\left(\frac{t_{WR}(m)}{t_{cyc}}, \frac{t_{DH}(m)}{t_{cyc}}\right) - 1$$

$$W_SETUP + W_STROBE + W_HOLD \geq \frac{t_{WC}(m)}{t_{cyc}} - 3$$

Figure 13. Timing Waveform of an ASRAM Write



3.1.3 Taking Into Account PCB Delays

The equations described in [Section 3.1.2](#) are for the ideal case, when board design does not contribute delays. Board characteristics, such as impedance, loading, length, number of nodes, etc., affect how the device driver behaves. Signals driven by the EMIF will be delayed when they reach the ASRAM and conversely. [Table 18](#) lists the delays shown in [Figure 14](#) and [Figure 15](#) due to PCB affects. The PCB delays are board specific and must be estimated or determined through the use of IBIS modeling. The signals denoted ($\overline{\text{ASRAM}}$) are the signals seen at the ASRAM. For example, $\overline{\text{EM_CS}}$ represents the signal at the EMIF and $\overline{\text{EM_CS}}(\text{ASRAM})$ represents the delayed signal seen at the ASRAM.

Table 18. ASRAM Timing Requirements With PCB Delays

| Parameter | Description |
|---------------------|---|
| Read Access | |
| $t_{\text{EM_CS}}$ | Delay on $\overline{\text{EM_CS}}$ from EMIF to ASRAM. $\overline{\text{EM_CS}}$ is driven by EMIF. |
| $t_{\text{EM_A}}$ | Delay on EM_A from EMIF to ASRAM. EM_A is driven by EMIF. |
| $t_{\text{EM_OE}}$ | Delay on $\overline{\text{EM_OE}}$ from EMIF to ASRAM. $\overline{\text{EM_OE}}$ is driven by EMIF. |
| $t_{\text{EM_D}}$ | Delay on EM_D from ASRAM to EMIF. EM_D is driven by ASRAM. |
| Write Access | |
| $t_{\text{EM_CS}}$ | Delay on $\overline{\text{EM_CS}}$ from EMIF to ASRAM. $\overline{\text{EM_CS}}$ is driven by EMIF. |
| $t_{\text{EM_A}}$ | Delay on EM_A from EMIF to ASRAM. EM_A is driven by EMIF. |
| $t_{\text{EM_WE}}$ | Delay on $\overline{\text{EM_WE}}$ from EMIF to ASRAM. $\overline{\text{EM_WE}}$ is driven by EMIF. |
| $t_{\text{EM_D}}$ | Delay on EM_D from EMIF to ASRAM. EM_D is driven by EMIF. |

From Figure 14, the following equations may be derived. t_{cyc} is the period at which the EMIF operates. The R_SETUP, R_STROBE, and R_HOLD fields are programmed in terms of EMIF cycles where as the data sheet specifications are typically given in nano seconds. This explains the presence of t_{cyc} in the denominator of the following equations. A minus 1 is included in the equations because each field in AnCR is programmed in terms of EMIF clock cycles, minus 1 cycle. For example, R_SETUP is equal to R_SETUP width in EMIF clock cycles minus 1 cycle.

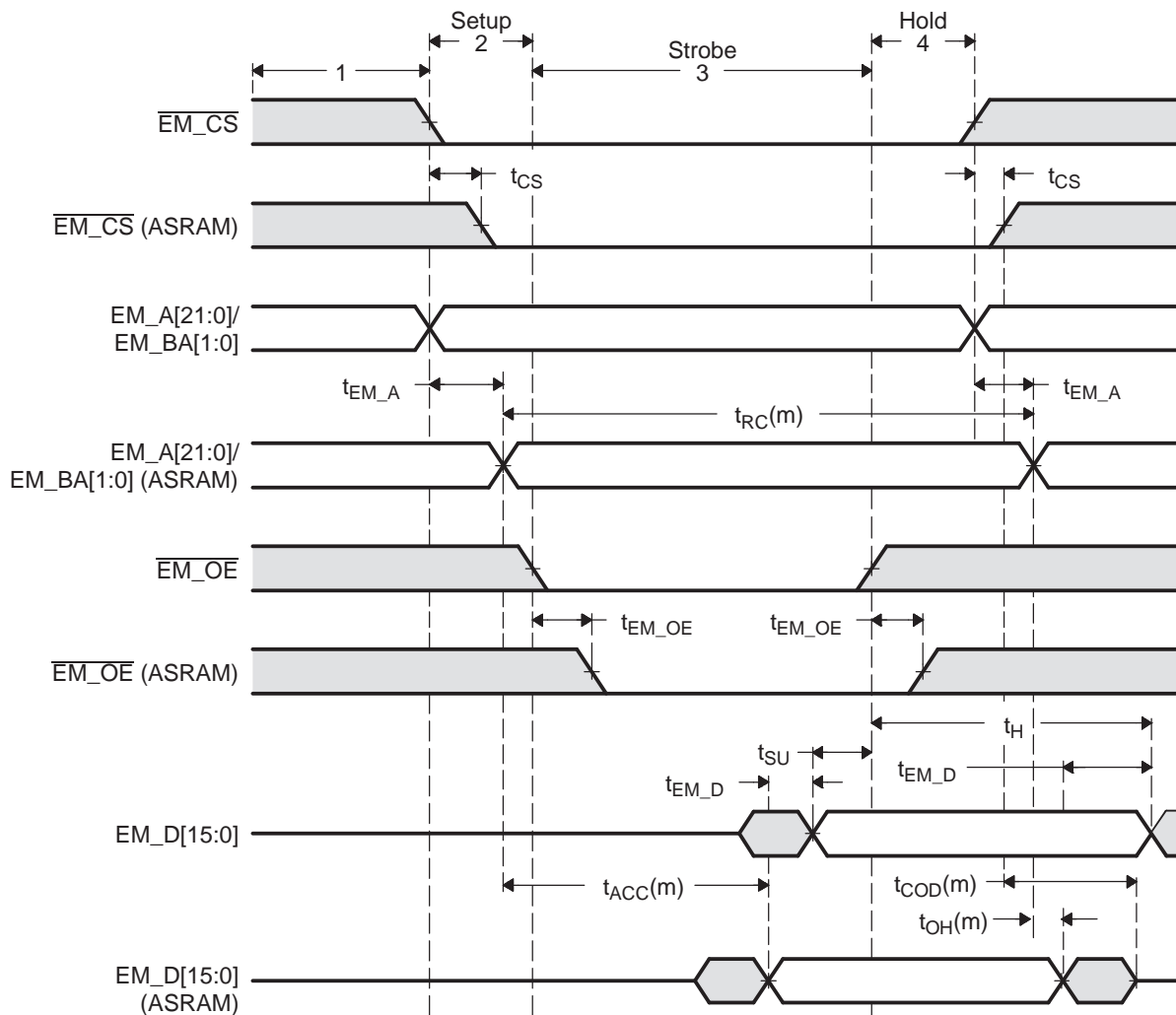
$$R_SETUP + R_STROBE \geq \frac{(t_{EM_A} + t_{ACC}(m) + t_{SU} + t_{EM_D})}{t_{cyc}} - 1$$

$$R_SETUP + R_STROBE + R_HOLD \geq \frac{t_{RC}(m)}{t_{cyc}} - 3$$

$$R_HOLD \geq \frac{(t_H - t_{EM_D} - t_{OH}(m) - t_{EM_A})}{t_{cyc}} - 1$$

$$TA \geq \frac{(t_{EM_CS} + t_{COD}(m) + t_{EM_D})}{t_{cyc}} - 1$$

Figure 14. Timing Waveform of an ASRAM Read with PCB Delays



From Figure 15, the following equations may be derived. t_{cyc} is the period at which the EMIF operates. The W_SETUP, W_STROBE, and W_HOLD fields are programmed in terms of EMIF cycles where as the data sheet specifications are typically given in nano seconds. This explains the presence of t_{cyc} in the denominator of the following equations. A minus 1 is included in the equations because each field in AnCR is programmed in terms of EMIF clock cycles, minus 1 cycle. For example, W_SETUP is equal to W_SETUP width in EMIF clock cycles minus 1 cycle.

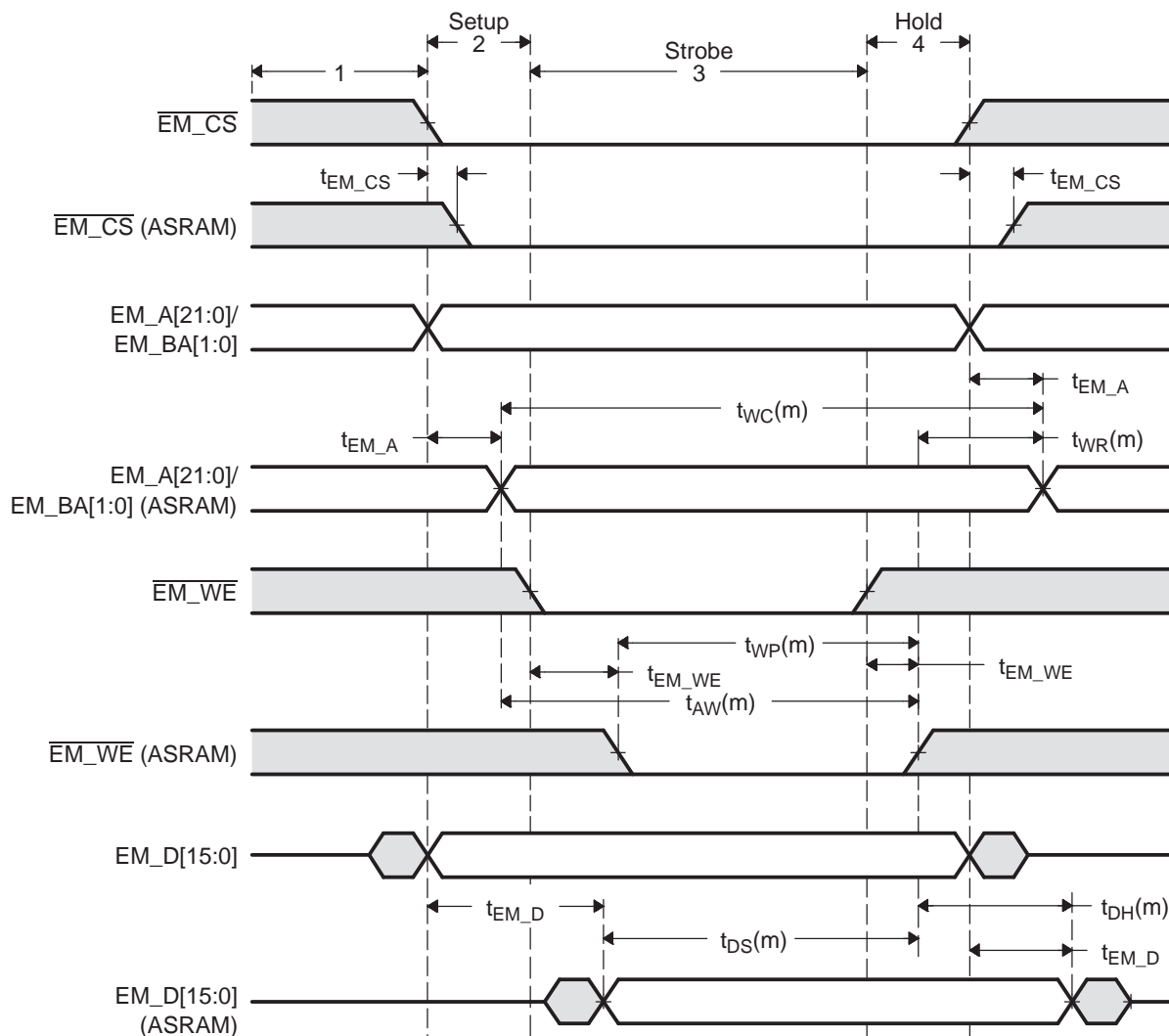
$$W_STROBE \geq \frac{t_{WP}(m)}{t_{cyc}} - 1$$

$$W_SETUP + W_STROBE \geq \max\left(\frac{(t_{EM_A} + t_{AW}(m) - t_{EM_WE})}{t_{cyc}}, \frac{(t_{EM_D} + t_{DS}(m) - t_{EM_WE})}{t_{cyc}}\right) - 1$$

$$W_HOLD \geq \max\left(\frac{(t_{EM_WE} + t_{WR}(m) - t_{EM_A})}{t_{cyc}}, \frac{(t_{EM_WE} + t_{DH}(m) - t_{EM_D})}{t_{cyc}}\right) - 1$$

$$W_SETUP + W_STROBE + W_HOLD \geq \frac{t_{WC}(m)}{t_{cyc}} - 3$$

Figure 15. Timing Waveform of an ASRAM Write with PCB Delays



3.1.4 Example Using TC5516100FT-12

This section takes you through the configuration steps required to implement Toshiba's TC55V1664FT-12 ASRAM with the EMIF. The following assumptions are made:

- ASRAM is connected to chip select space 3 ($\overline{\text{EM_CS}}[3]$)
- EMIF clock speed is 100 MHz ($t_{\text{cyc}} = 10 \text{ nS}$)

Table 19 lists the data sheet specifications for the EMIF and Table 20 lists the data sheet specifications for the ASRAM.

Table 19. EMIF Timing Requirements for TC5516100FT-12 Example

| Parameter | Description | Min | Max | Units |
|-----------------|--|-----|-----|-------|
| t_{SU} | Data Setup time, data valid before $\overline{\text{EM_OE}}$ high | 5 | | nS |
| t_{H} | Data Hold time, data valid after $\overline{\text{EM_OE}}$ high | 0 | | nS |

Table 20. ASRAM Timing Requirements for TC5516100FT-12 Example

| Parameter | Description | Min | Max | Units |
|------------------|--|-----|-----|-------|
| t_{ACC} | Address Access time | | 12 | nS |
| t_{OH} | Output data Hold time for address change | 3 | | nS |
| t_{RC} | Read cycle time | 12 | | nS |
| t_{WP} | Write Pulse width | 8 | | nS |
| t_{AW} | Address valid to end of Write | 9 | | nS |
| t_{DS} | Data Setup time | 7 | | nS |
| t_{WR} | Write Recovery time | 0 | | nS |
| t_{DH} | Data Hold time | 0 | | nS |
| t_{WC} | Write Cycle time | 12 | | nS |
| t_{COD} | Output Disable time from chip enable | | 7 | |

Table 21 lists the values of the PCB board delays. The delays were estimated using the rule that there is 180 pS of delay for every 1 inch of trace.

Table 21. Measured PCB Delays for TC5516100FT-12 Example

| Parameter | Description | Delay (ns) |
|---------------------|---|------------|
| Read Access | | |
| $t_{\text{EM_CS}}$ | Delay on $\overline{\text{EM_CS}}$ from EMIF to ASRAM. $\overline{\text{EM_CS}}$ is driven by EMIF. | 0.36 |
| $t_{\text{EM_A}}$ | Delay on EM_A from EMIF to ASRAM. EM_A is driven by EMIF. | 0.27 |
| $t_{\text{EM_OE}}$ | Delay on $\overline{\text{EM_OE}}$ from EMIF to ASRAM. $\overline{\text{EM_OE}}$ is driven by EMIF. | 0.36 |
| $t_{\text{EM_D}}$ | Delay on EM_D from ASRAM to EMIF. EM_D is driven by ASRAM. | 0.45 |
| Write Access | | |
| $t_{\text{EM_CS}}$ | Delay on $\overline{\text{EM_CS}}$ from EMIF to ASRAM. $\overline{\text{EM_CS}}$ is driven by EMIF. | 0.36 |
| $t_{\text{EM_A}}$ | Delay on EM_A from EMIF to ASRAM. EM_A is driven by EMIF. | 0.27 |
| $t_{\text{EM_WE}}$ | Delay on $\overline{\text{EM_WE}}$ from EMIF to ASRAM. $\overline{\text{EM_WE}}$ is driven by EMIF. | 0.36 |
| $t_{\text{EM_D}}$ | Delay on EM_D from EMIF to ASRAM. EM_D is driven by EMIF. | 0.45 |

Inserting these values into the equations defined above allows you to determine the values for SETUP, STROBE, HOLD, and TA. For a read:

$$R_SETUP + R_STROBE \geq \frac{(t_{EM_A} + t_{ACC}(m) + t_{SU} + t_{EM_D})}{t_{cyc}} - 1 \geq \frac{(0.27 + 12 + 5 + 0.45)}{10} - 1 \geq 0.78$$

$$R_SETUP + R_STROBE + R_HOLD \geq \frac{t_{RC}(m)}{t_{cyc}} - 3 \geq \left(\frac{12}{10}\right) - 3 \geq -1.8$$

$$R_HOLD \geq \frac{(t_H - t_{EM_D} - t_{OH}(m) - t_{EM_A})}{t_{cyc}} - 1 \geq \frac{(0 - 0.45 - 3 - 0.27)}{10} - 1 \geq -1.37$$

$$TA \geq \frac{(t_{EM_CS} + T_{COD}(m) + t_{EM_D})}{t_{cyc}} - 1 \geq \frac{(0.36 + 7 + 0.45)}{10} - 1 \geq -0.22$$

Therefore if $R_SETUP = 0$, then $R_STROBE = 0$, $R_HOLD = 0$, and $TA = 0$.

For a write:

$$W_STROBE \geq \frac{t_{WP}(m)}{t_{cyc}} - 1 \geq \left(\frac{8}{10}\right) - 1 \geq -0.2$$

$$\begin{aligned} W_SETUP + W_STROBE &\geq \max\left(\frac{(t_{EM_A} + t_{AW}(m) - t_{EM_WE})}{t_{cyc}}, \frac{(t_{EM_D} + t_{DS}(m) - t_{EM_WE})}{t_{cyc}}\right) - 1 \\ &\geq \max\left(\frac{(0.36 + 0 - 0.27)}{10}, \frac{(0.36 + 0 - 0.45)}{10}\right) - 1 \geq -0.92 \end{aligned}$$

$$\begin{aligned} W_HOLD &\geq \max\left(\frac{(t_{EM_WE} + t_{WR}(m) - t_{EM_A})}{t_{cyc}}, \frac{(t_{EM_WE} + t_{DH}(m) - t_{EM_D})}{t_{cyc}}\right) - 1 \\ &\geq \max\left(\frac{(0.27 + 9 - 0.36)}{10}, \frac{(0.45 + 7 - 0.36)}{10}\right) - 1 \geq -0.1 \end{aligned}$$

$$W_SETUP + W_STROBE + W_HOLD \geq \frac{t_{WC}(m)}{t_{cyc}} - 3 \geq \left(\frac{12}{10}\right) - 3 \geq -1.8$$

Therefore, $W_SETUP = 0$, $W_STROBE = 0$, and $W_HOLD = 0$.

Since the value of the W_SETUP/R_SETUP, W_STROBE/R_STROBE, W_HOLD/R_HOLD, and TA fields are equal to EMIF clock cycles minus 1 cycle, the A2CR should be configured as in [Table 22](#). In this example, the EM_WAIT signal is not implemented; therefore, the asynchronous wait cycle configuration register (AWCCR) does not need to be programmed.

Table 22. Configuring A2CR for TC5516100FT-12 Example

| Parameter | Setting |
|-------------------|---|
| SS | Select Strobe mode. <ul style="list-style-type: none"> SS = 0. Places EMIF in Normal Mode. |
| EW | Extended Wait mode enable. <ul style="list-style-type: none"> EW = 0. Disabled Extended wait mode. |
| W_SETUP/R_SETUP | Read/Write setup widths. <ul style="list-style-type: none"> W_SETUP = 0 R_SETUP = 0 |
| W_STROBE/R_STROBE | Read/Write strobe widths. <ul style="list-style-type: none"> W_STROBE = 0 R_STROBE = 0 |
| W_HOLD/R_HOLD | Read/Write hold widths. <ul style="list-style-type: none"> W_HOLD = 0 R_HOLD = 0 |
| TA | Minimum turnaround time. <ul style="list-style-type: none"> TA = 0 |
| ASIZE | Asynchronous Device Bus Width. <ul style="list-style-type: none"> ASIZE = 1, select a 16-bit data bus width |

3.2 Interfacing to NAND Flash

The following example explains how to interface the EMIF to the Hynix HY27UA081G1M NAND Flash device. [Section 2.5.6.2](#) describes how to connect the EMIF to the HY27UA081G1M.

3.2.1 Margin Requirements

The Flash interface is typically a low-performance interface compared to synchronous memory interfaces, high-speed asynchronous memory interfaces, and high-speed FIFO interfaces. For this reason, this example gives little attention to minimizing the amount of margin required when programming the asynchronous timing parameters. The approach used requires approximately 10 ns of margin on all parameters, which is not significant for a 100-ns read or write cycle. For additional details on minimizing the amount of margin, see the ASRAM example given in [Section 3.1](#).

Table 23. Recommended Margins

| Timing Parameter | Recommended Margin |
|------------------|--------------------|
| Output Setup | 10 nS |
| Output Hold | 10 nS |
| Input Setup | 10 nS |
| Input Hold | 10 nS |

3.2.2 Meeting AC Timing Requirements for NAND Flash

When configuring the EMIF to interface to NAND Flash, you must consider the AC timing requirements of the NAND Flash as well as the AC timing requirements of the EMIF. These can be found in the data sheet for each respective device. The read and write asynchronous cycles are programmed separately in the asynchronous configuration register (AnCR).

As described in [Section 2.5.6](#), a NAND Flash access cycle is composed of a command, address, and data phases. The EMIF 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 EMIF. The command and address phases of a NAND Flash access cycle are asynchronous writes performed by the EMIF where as the data phase can be either an asynchronous write or a read depending on whether the NAND Flash is being programmed or read.

Therefore, to determine the required EMIF configuration to interface to the NAND Flash for a read operation, [Table 24](#) and [Table 25](#) list the AC timing parameters that must be considered.

Table 24. EMIF Read Timing Requirements

| Parameter | Description |
|-----------|---|
| t_{SU} | Data Setup time, data valid before $\overline{EM_OE}$ high |
| t_H | Data Hold time, data valid after $\overline{EM_OE}$ high |

Table 25. NAND Flash Read Timing Requirements

| Parameter | Description |
|-----------|--------------------------------------|
| t_{RP} | Read Pulse width |
| t_{REA} | Read Enable Access time |
| t_{CEA} | Chip Enable low to output valid |
| t_{CHZ} | Chip Enable high to output High-Z |
| t_{RC} | Read Cycle time |
| t_{RHZ} | Read enable high to output High-Z |
| t_{CLR} | Command Latch low to Read enable low |

[Figure 16](#) shows an asynchronous read access and describes how the EMIF and NAND Flash AC timing requirements work together to define the values for R_SETUP, R_STROBE, and R_HOLD.

From Figure 16, the following equations may be derived. t_{cyc} is the period at which the EMIF operates. The R_SETUP, R_STROBE, and R_HOLD fields are programmed in terms of EMIF cycles where as the data sheet specifications are typically given in nano seconds. This explains the presence of t_{cyc} in the denominator of the following equations. A minus 1 is included in the equations because each field in AnCR is programmed in terms of EMIF clock cycles, minus 1 cycle. For example, R_SETUP is equal to R_SETUP width in EMIF clock cycles minus 1 cycle.

$$R_SETUP \geq \frac{t_{CLR}(m)}{t_{cyc}} - 1$$

$$R_STROBE \geq \max\left(\frac{(t_{REA}(m) + t_{SU})}{t_{cyc}}, \frac{t_{RP}(m)}{t_{cyc}}\right) - 1$$

$$R_SETUP + R_STROBE \geq \frac{(t_{CEA}(m) + t_{SU})}{t_{cyc}} - 1$$

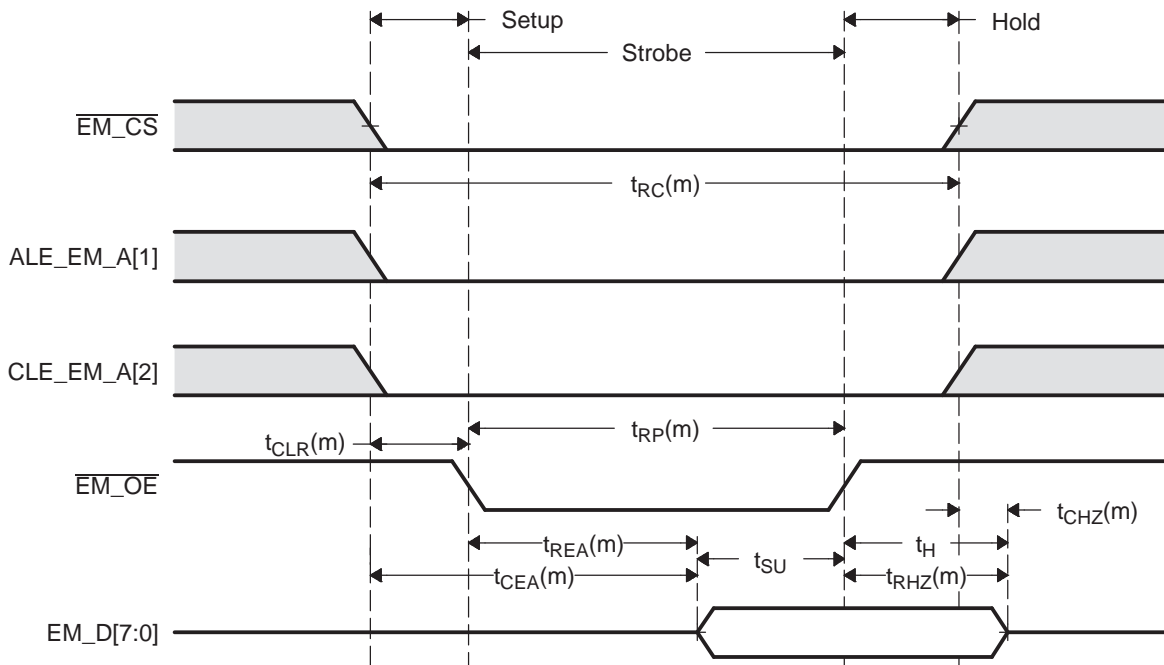
$$R_HOLD \geq \frac{(t_H - t_{CHZ}(m))}{t_{cyc}} - 1$$

$$R_SETUP + R_STROBE + R_HOLD \geq \frac{t_{RC}(m)}{t_{cyc}} - 3$$

The EMIF offers an additional parameter, TA, that defines the turnaround time between read and write cycles. This parameter protects against the situation when the output turn-off time of the memory is longer than the time it takes to start the next write cycle. If this is the case, the EMIF will drive data at the same time as the memory, causing contention on the bus. By examining Figure 16, the equation for TA can be derived as:

$$TA \geq \max\left(\frac{t_{CHZ}(m)}{t_{cyc}}, \frac{t_{RHZ}(m) - (R_HOLD + 1)t_{cyc}}{t_{cyc}}\right) - 1$$

Figure 16. Timing Waveform of a NAND Flash Read



To determine the required EMIF configuration to interface to the NAND Flash for a write operation, [Table 26](#) lists the NAND AC timing parameters for a command latch, address latch, and data input latch that must be considered.

Table 26. NAND Flash Write Timing Requirements

| Parameter | Description |
|-----------|----------------------------|
| t_{WP} | Write Pulse width |
| t_{CLS} | CLE Setup time |
| t_{ALS} | ALE Setup time |
| t_{CS} | \overline{CS} Setup time |
| t_{DS} | Data Setup time |
| t_{CLH} | CLE Hold time |
| t_{ALH} | ALE Hold time |
| t_{CH} | \overline{CS} Hold time |
| t_{DH} | Data Hold time |
| t_{WC} | Write Cycle time |

[Figure 17](#) to [Figure 19](#) show the command latch, address latch, and data input latch of the NAND access.

From [Figure 17](#) to [Figure 19](#), the following equations may be derived. t_{cyc} is the period at which the EMIF operates. The W_SETUP, W_STROBE, and W_HOLD fields are programmed in terms of EMIF cycles where as the data sheet specifications are typically given in nano seconds. This explains the presence of t_{cyc} in the denominator of the following equations. A minus 1 is included in the equations because each field in AnCR is programmed in terms of EMIF clock cycles, minus 1 cycle. For example, W_SETUP is equal to W_SETUP width in EMIF clock cycles minus 1 cycle.

$$W_SETUP \geq \max\left(\frac{t_{CLS}(m)}{t_{cyc}}, \frac{t_{ALS}(m)}{t_{cyc}}, \frac{t_{CS}(m)}{t_{cyc}}\right) - 1$$

$$W_STROBE \geq \frac{t_{WP}(m)}{t_{cyc}} - 1$$

$$W_SETUP + W_STROBE \geq \frac{t_{DS}(m)}{t_{cyc}} - 1$$

$$W_HOLD \geq \max\left(\frac{t_{CLH}(m)}{t_{cyc}}, \frac{t_{ALH}(m)}{t_{cyc}}, \frac{t_{CH}(m)}{t_{cyc}}, \frac{t_{DH}(m)}{t_{cyc}}\right) - 1$$

$$W_SETUP + W_STROBE + W_HOLD \geq \frac{t_{WC}(m)}{t_{cyc}} - 3$$

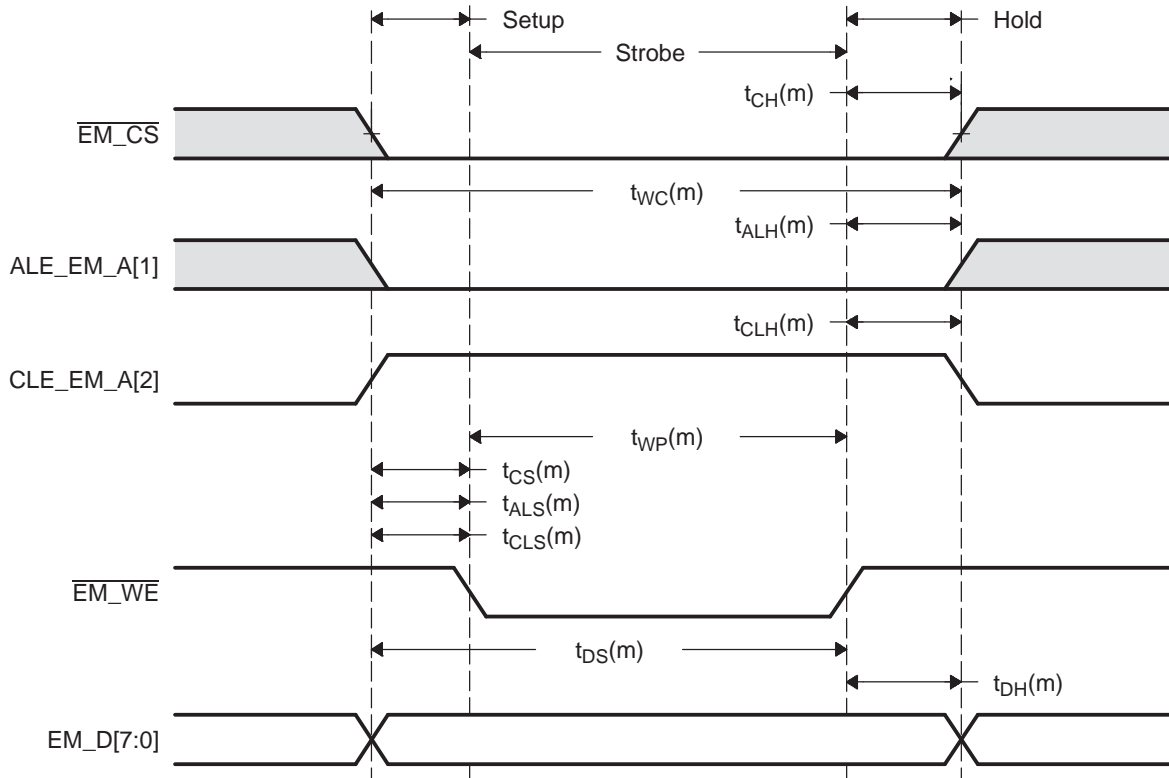
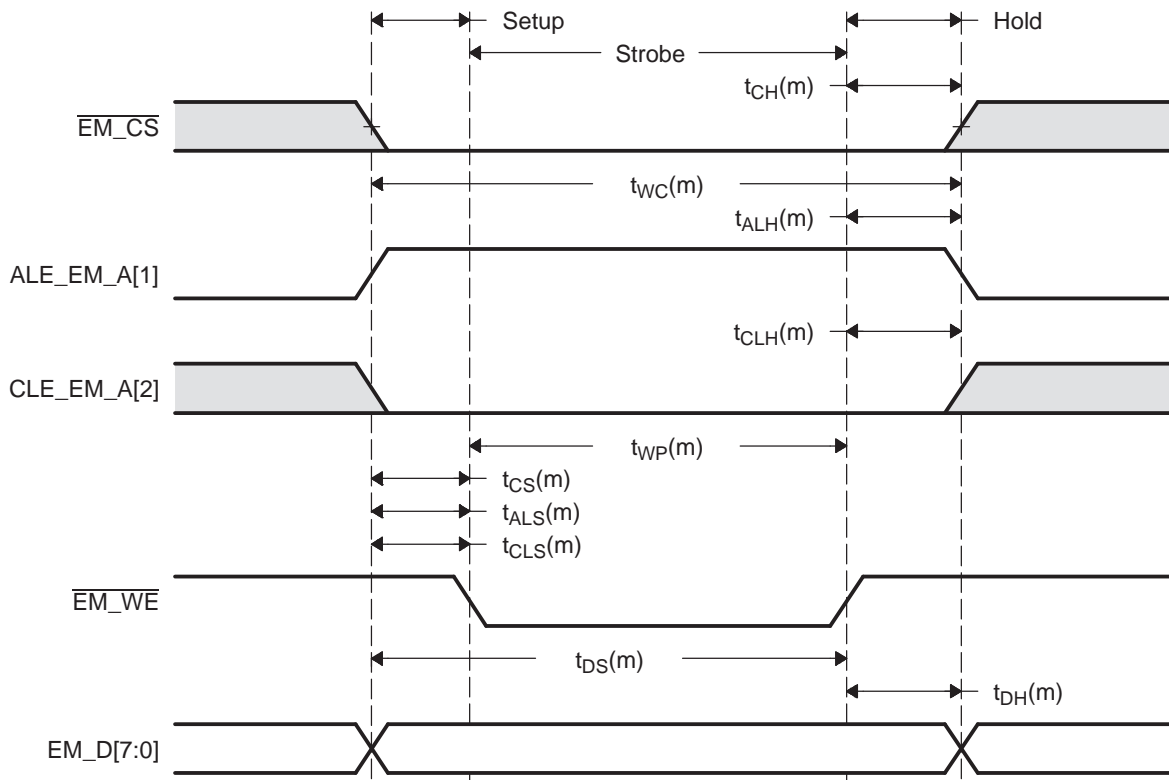
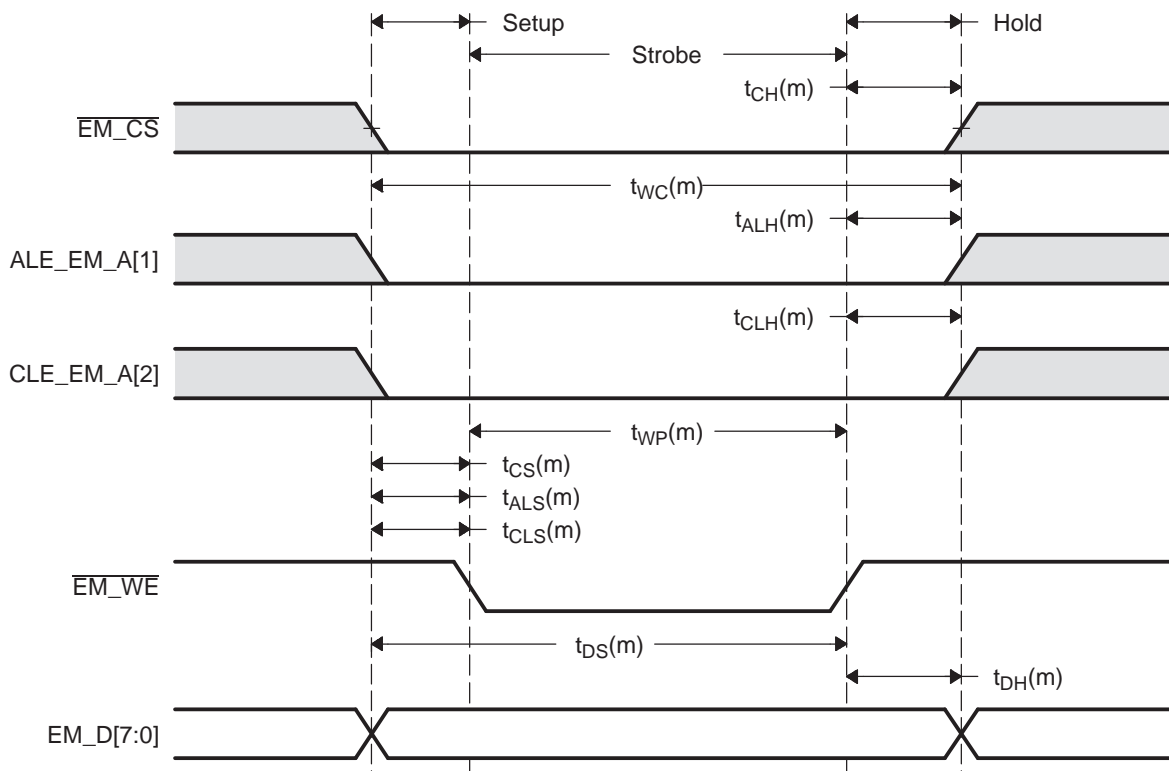
Figure 17. Timing Waveform of a NAND Flash Command Write

Figure 18. Timing Waveform of a NAND Flash Address Write


Figure 19. Timing Waveform of a NAND Flash Data Write



3.2.3 Example Using Hynix HY27UA081G1M

This section takes you through the configuration steps required to implement Hynix's HY27UA081G1M NAND Flash with the EMIF. The following assumptions are made:

- NAND Flash is connected to chip select space 2 ($\overline{\text{EM_CS}}[2]$)
- EMIF clock speed is 100 MHz ($t_{\text{cyc}} = 10 \text{ nS}$)

[Table 27](#) lists the data sheet specifications for the EMIF and [Table 28](#) lists the data sheet specifications for the NAND Flash.

Table 27. EMIF Timing Requirements for HY27UA081G1M Example

| Parameter | Description | Min | Max | Units |
|-----------------|--|-----|-----|-------|
| t_{SU} | Data Setup time, data valid before $\overline{\text{EM_OE}}$ high | 5 | | nS |
| t_{H} | Data Hold time, data valid after $\overline{\text{EM_OE}}$ high | 0 | | nS |

Table 28. NAND Flash Timing Requirements for HY27UA081G1M Example

| Parameter | Description | Min | Max | Units |
|------------------|--------------------------------------|-----|-----|-------|
| t_{RP} | Read Pulse width | 60 | | nS |
| t_{REA} | Read Enable Access time | | 60 | nS |
| t_{CEA} | Chip Enable low to output valid | | 75 | nS |
| t_{CHZ} | Chip Enable high to output High-Z | | 20 | nS |
| t_{RC} | Read Cycle time | 80 | | nS |
| t_{RHZ} | Read Enable high to output High-Z | | 30 | nS |
| t_{CLR} | Command Latch low to Read enable low | 10 | | nS |
| t_{WP} | Write Pulse width | 60 | | nS |
| t_{CLS} | CLE Setup time | 0 | | nS |
| t_{ALS} | ALE Setup time | 0 | | nS |
| t_{CS} | $\overline{\text{CS}}$ Setup time | 0 | | nS |
| t_{DS} | Data Setup time | 20 | | nS |
| t_{CLH} | CLE Hold time | 10 | | nS |
| t_{ALH} | ALE Hold time | 10 | | nS |
| t_{CH} | $\overline{\text{CS}}$ Hold time | 10 | | nS |
| t_{DH} | Data Hold time | 10 | | nS |
| t_{WC} | Write Cycle time | 80 | | nS |

Inserting these values into the equations defined above allows you to determine the values for SETUP, STROBE, HOLD, and TA. For a read:

$$R_SETUP \geq \frac{t_{CLR}(m)}{t_{cyc}} - 1 \geq \left(\frac{10}{10}\right) - 1 \geq 0$$

$$R_STROBE \geq \max\left(\frac{(t_{REA}(m) + t_{SU})}{t_{cyc}}, \frac{t_{RP}}{t_{cyc}}\right) - 1 \geq \left(\frac{65}{10}\right) - 1 \geq 5.5$$

$$R_SETUP + R_STROBE \geq \frac{(t_{CEA} + t_{SU})}{t_{cyc}} - 1 \geq \frac{(75 + 5)}{10} - 1 \geq 7$$

$$R_HOLD \geq \frac{(t_H - t_{CHZ}(m))}{t_{cyc}} - 1 \geq \frac{(0 - 20)}{10} - 1 \geq -3$$

$$R_SETUP + R_STROBE + R_HOLD \geq \frac{t_{RC}(m)}{t_{cyc}} - 3 \geq \left(\frac{80}{10}\right) - 3 \geq 5$$

Therefore with a 10 nS margin added in, $R_SETUP \geq 1.0$, $R_STROBE \geq 6.5$, and $R_HOLD \geq 0$.

After solving for R_HOLD , TA may be calculated:

$$TA \geq \max\left(\frac{t_{CHZ}(m)}{t_{cyc}}, \frac{t_{RHZ}(m) - (R_HOLD + 1)t_{cyc}}{t_{cyc}}\right) - 1 \geq \left(\frac{20}{10}\right) - 1 \geq 1$$

Adding a 10 ns margin, $TA \geq 2$.

For a write:

$$W_STROBE \geq \frac{t_{WP}(m)}{t_{cyc}} - 1 \geq \left(\frac{60}{10}\right) - 1 \geq 5$$

$$W_SETUP \geq \max\left(\frac{t_{CLS}(m)}{t_{cyc}}, \frac{t_{ALS}(m)}{t_{cyc}}, \frac{t_{CS}(m)}{t_{cyc}}\right) - 1 \geq \left(\frac{0}{10}\right) - 1 \geq -1$$

$$W_SETUP + W_STROBE \geq \frac{t_{DS}(m)}{t_{cyc}} - 1 \geq \left(\frac{20}{10}\right) - 1 \geq 1$$

$$W_HOLD \geq \max\left(\frac{t_{CLH}(m)}{t_{cyc}}, \frac{t_{ALH}(m)}{t_{cyc}}, \frac{t_{CH}(m)}{t_{cyc}}, \frac{t_{DH}(m)}{t_{cyc}}\right) - 1 \geq \left(\frac{10}{10}\right) - 1 \geq 0$$

$$W_SETUP + W_STROBE + W_HOLD \geq \frac{t_{WC}(m)}{t_{cyc}} - 3 \geq \left(\frac{80}{10}\right) - 3 \geq 5$$

Therefore with a 10 nS margin added in, $W_SETUP \geq 0$, $W_STROBE \geq 6$, and $W_HOLD \geq 1$.

Since the value of the W_SETUP/R_SETUP, W_STROBE/R_STROBE, W_HOLD/R_HOLD, and TA fields are equal to EMIF clock cycles minus 1 cycle, the A1CR should be configured as in [Table 29](#). In this example, although the EM_WAIT signal is connected to the R/B signal of the NAND Flash the Extended Wait mode of the EMIF is not used, therefore the asynchronous wait cycle configuration register (AWCCR) does not need to be programmed.

Table 29. Configuring A1CR for HY27UA081G1M Example

| Parameter | Setting |
|-------------------|--|
| SS | Select Strobe mode. <ul style="list-style-type: none"> SS = 0. Places EMIF in Normal Mode. |
| EW | Extended Wait mode enable. <ul style="list-style-type: none"> EW = 0. Disabled Extended wait mode. |
| W_SETUP/R_SETUP | Read/Write setup widths. <ul style="list-style-type: none"> W_SETUP = 0 R_SETUP = 2 |
| W_STROBE/R_STROBE | Read/Write strobe widths. <ul style="list-style-type: none"> W_STROBE = 6 R_STROBE = 7 |
| W_HOLD/R_HOLD | Read/Write hold widths. <ul style="list-style-type: none"> W_HOLD = 1 R_HOLD = 0 |
| TA | Minimum turnaround time. <ul style="list-style-type: none"> TA = 2 |
| ASIZE | Asynchronous device bus width. <ul style="list-style-type: none"> ASIZE = 0, select an 8-bit data bus width. |

Since this is a NAND Flash example, the EMIF must be configured for NAND Flash mode. This is accomplished by configuring the NAND Flash control register (NANDFCR) as in [Table 30](#). In NANDFCR, chip select space 2 must be configured with NAND Flash mode enabled.

Table 30. Configuring NANDFCR for HY27UA081G1M Example

| Parameter | Setting |
|-----------|--|
| CS5ECC | NAND Flash ECC start for chip select 5. <ul style="list-style-type: none"> CS5ECC = 0. Not set during configuration. Only set just prior to reading or writing data. |
| CS4ECC | NAND Flash ECC start for chip select 4. <ul style="list-style-type: none"> CS4ECC = 0. Not set during configuration. Only set just prior to reading or writing data. |
| CS3ECC | NAND Flash ECC start for chip select 3. <ul style="list-style-type: none"> CS3ECC = 0. Not set during configuration. Only set just prior to reading or writing data. |
| CS2ECC | NAND Flash ECC start for chip select 2. <ul style="list-style-type: none"> CS2ECC = 0. Not set during configuration. Only set just prior to reading or writing data. |
| CS5NAND | NAND Flash mode for chip select 5. <ul style="list-style-type: none"> CS5NAND = 0. NAND Flash mode is disabled. |
| CS4NAND | NAND Flash mode for chip select 4. <ul style="list-style-type: none"> CS4NAND = 0. NAND Flash mode is disabled. |
| CS3NAND | NAND Flash mode for chip select 3. <ul style="list-style-type: none"> CS3NAND = 0. NAND Flash mode is disabled. |
| CS2NAND | NAND Flash mode for chip select 2. <ul style="list-style-type: none"> CS5NAND = 1. NAND Flash mode is enabled. |

4 Registers

The external memory interface (EMIF) is controlled by programming its internal memory-mapped registers (MMRs). [Table 31](#) lists the memory-mapped registers for the EMIF. See the device-specific data manual for the memory address of these registers. All other register offset addresses not listed in [Table 31](#) should be considered as reserved locations and the register contents should not be modified.

Note: The EMIF MMRs only support word (4 byte) accesses. Performing a byte (8 bit) or halfword (16 bit) write to a register results in unknown behavior.

Table 31. External Memory Interface (EMIF) Registers

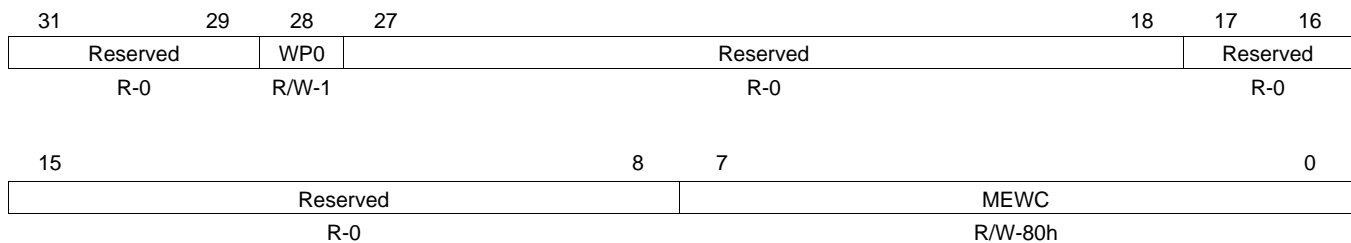
| Offset | Acronym | Register Description | Section |
|--------|-----------|---|-----------------------------|
| 04h | AWCCR | Asynchronous Wait Cycle Configuration Register | Section 4.1 |
| 10h | A1CR | Asynchronous 1 Configuration Register (CS2 space) | Section 4.2 |
| 14h | A2CR | Asynchronous 2 Configuration Register (CS3 space) | Section 4.2 |
| 18h | A3CR | Asynchronous 3 Configuration Register (CS4 space) | Section 4.2 |
| 1Ch | A4CR | Asynchronous 4 Configuration Register (CS5 space) | Section 4.2 |
| 40h | EIRR | EMIF Interrupt Raw Register | Section 4.3 |
| 44h | EIMR | EMIF Interrupt Mask Register | Section 4.4 |
| 48h | EIMSR | EMIF Interrupt Mask Set Register | Section 4.5 |
| 4Ch | EIMCR | EMIF Interrupt Mask Clear Register | Section 4.6 |
| 60h | NANDFCR | NAND Flash Control Register | Section 4.7 |
| 64h | NANDFSR | NAND Flash Status Register | Section 4.8 |
| 70h | NANDF1ECC | NAND Flash 1 ECC Register (CS2 Space) | Section 4.9 |
| 74h | NANDF2ECC | NAND Flash 2 ECC Register (CS3 Space) | Section 4.9 |
| 78h | NANDF3ECC | NAND Flash 3 ECC Register (CS4 Space) | Section 4.9 |
| 7Ch | NANDF4ECC | NAND Flash 4 ECC Register (CS5 Space) | Section 4.9 |

4.1 Asynchronous Wait Cycle Configuration Register (AWCCR)

The asynchronous wait cycle configuration register (AWCCR) is used to configure the parameters for extended wait cycles. Both the polarity of the EM_WAIT pin and the maximum allowable number of extended wait cycles can be configured. The AWCCR is shown in Figure 20 and described in Table 32.

Note: The EW bit in the asynchronous configuration register (AnCR) must be set to allow for the insertion of extended wait cycles.

Figure 20. Asynchronous Wait Cycle Configuration Register (AWCCR)



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

Table 32. Asynchronous Wait Cycle Configuration Register (AWCCR) Field Descriptions

| Bit | Field | Value | Description |
|-------|----------|--------|--|
| 31-29 | Reserved | 0 | Reserved |
| 28 | WP0 | 0 1 | WAIT polarity bit. This bit defines the polarity of the EM_WAIT pin. Insert wait cycles if EM_WAIT is low. Insert wait cycles if EM_WAIT is high. |
| 27-18 | Reserved | 0 | Reserved |
| 17-16 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0. |
| 15-8 | Reserved | 0 | Reserved |
| 7-0 | MEWC | 0-FFh | Maximum extended wait cycles. The EMIF will wait for a maximum of (MEWC + 1) 16 clock cycles before it stops inserting asynchronous wait cycles and proceeds to the hold period of the access. |

4.2 Asynchronous *n* Configuration Registers (A1CR-A4CR)

The asynchronous configuration register (*AnCR*) is used to configure the shaping of the address and control signals during an access to asynchronous memory. 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 *AnCR* is shown in [Figure 21](#) and described in [Table 33](#). There are four *AnCR*s. Each chip select space has a dedicated *AnCR*. This allows each chip select space to be programmed independently to interface to different asynchronous memory types.

Figure 21. Asynchronous *n* Configuration Register (*AnCR*)

| | | | | | | | | | |
|-------------------------|-------------------|-------------------------|--------|-------------------------|---------|----|--------|---|-------|
| 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; -*n* = value after reset

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

B. The W_STROBE and R_STROBE bits must not be cleared to 0 when operating in Extended Wait mode.

Table 33. Asynchronous *n* Configuration Register (*AnCR*) Field Descriptions

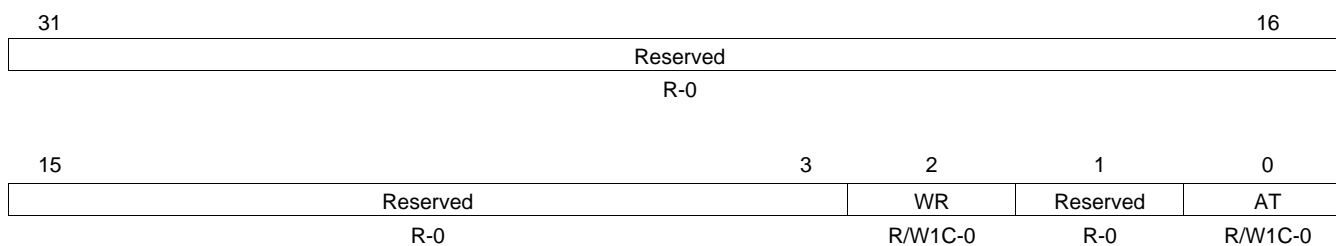
| Bit | Field | Value | Description |
|-------|----------|--------|--|
| 31 | SS | 0 1 | Select Strobe bit. This bit defines whether the asynchronous interface operates in Normal mode or Select Strobe mode. See Section 2.5 for details on the two modes of operation. Normal mode is enabled. Select Strobe mode is enabled. |
| 30 | EW | 0 1 | Extend Wait bit. This bit defines whether extended wait cycles will be enabled. See Section 2.5.8 on extended wait cycles for details. This bit field must be cleared to 0, if the EMIF on your device does not have a EM_WAIT pin. Extended wait cycles are disabled. Extended wait cycles are enabled. |
| 29-26 | W_SETUP | 0-Fh | Write setup width in EMIF clock cycles, minus 1 cycle. See Section 2.5.3 for details. |
| 25-20 | W_STROBE | 0-3Fh | Write strobe width in EMIF clock cycles, minus 1 cycle. See Section 2.5.3 for details. |
| 19-17 | W_HOLD | 0-7h | Write hold width in EMIF clock cycles, minus 1 cycle. See Section 2.5.3 for details. |
| 16-13 | R_SETUP | 0-Fh | Read setup width in EMIF clock cycles, minus 1 cycle. See Section 2.5.3 for details. |
| 12-7 | R_STROBE | 0-3Fh | Read strobe width in EMIF clock cycles, minus 1 cycle. See Section 2.5.3 for details. |
| 6-4 | R_HOLD | 0-7h | Read hold width in EMIF clock cycles, minus 1 cycle. See Section 2.5.3 for details. |
| 3-2 | TA | 0-3h | Minimum Turn-Around time. This field defines the minimum number of EMIF clock cycles between the end of one asynchronous access and the start of another, minus 1 cycle. This delay is not incurred by a read followed by a read or a write followed by a write to the same CS space. See Section 2.5.3 for details. |

Table 33. Asynchronous n Configuration Register (AnCR) Field Descriptions (continued)

| Bit | Field | Value | Description |
|-----|-------|-------|--|
| 1-0 | ASIZE | 0-3h | Asynchronous data bus width. This bit defines the width of the asynchronous device's data bus. |
| | | 0 | 8-bit data bus |
| | | 1h | 16-bit data bus |
| | | 2h-3h | Reserved |

4.3 EMIF Interrupt Raw Register (EIRR)

The EMIF interrupt raw register (EIRR) is used to monitor and clear the EMIF's hardware-generated interrupts. The bits in EIRR will be set when an interrupt condition occurs regardless of the status of the EMIF interrupt mask set register (EIMSR) and EMIF interrupt mask clear register (EIMCR). Writing a 1 to these bit fields will clear them as well as the corresponding bit field in the EMIF interrupt mask register (EIMR). The EIRR is shown in [Figure 22](#) and described in [Table 34](#).

Figure 22. EMIF Interrupt Raw Register (EIRR)


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

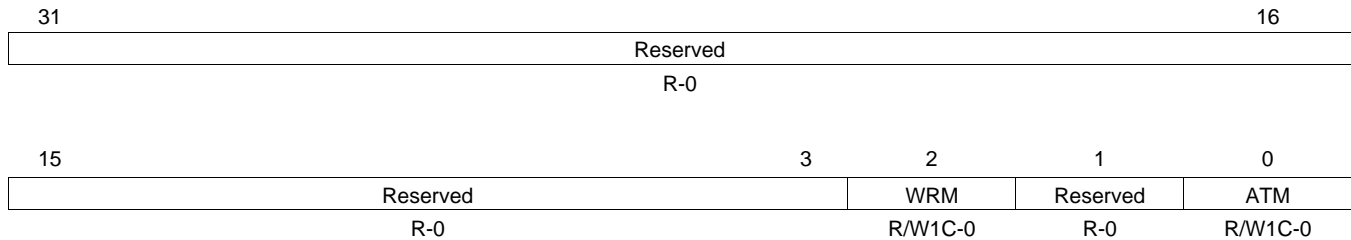
Table 34. EMIF Interrupt Raw Register (EIRR) Field Descriptions

| 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 | 0 | Wait Rise. This bit is set to 1 by hardware to indicate that a rising edge on the EM_WAIT pin has occurred. |
| | | 1 | Indicates that a rising edge has occurred on the EM_WAIT pin. Writing a 1 will clear this bit and the WRM bit in the EMIF interrupt mask register (EIMR). |
| 1 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0. |
| 0 | AT | 0 | Asynchronous Timeout. This bit is set to 1 by hardware to indicate that during an extended asynchronous memory access cycle the EM_WAIT pin did not go inactive within the number of cycles defined by the MEWC field in the asynchronous wait cycle configuration register (AWCCR). |
| | | 1 | Indicates that an asynchronous timeout has occurred. Writing a 1 will clear this bit and the ATM bit in the EMIF interrupt mask register (EIMR). |

4.4 EMIF Interrupt Mask Register (EIMR)

Like the EMIF interrupt raw register (EIRR), the EMIF interrupt mask register (EIMR) is used to monitor and clear the status of the EMIF's hardware-generated interrupts. The main difference between the two registers is that when the bit fields in EIMR are set, an active-high pulse will be sent to the CPU interrupt controller. Also, the bit fields in EIMR are only set to 1 if the associated interrupt has been enabled in the EMIF interrupt set register (EISR). The EIMR is shown in [Figure 23](#) and described in [Table 35](#).

Figure 23. EMIF Interrupt Mask Register (EIMR)



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

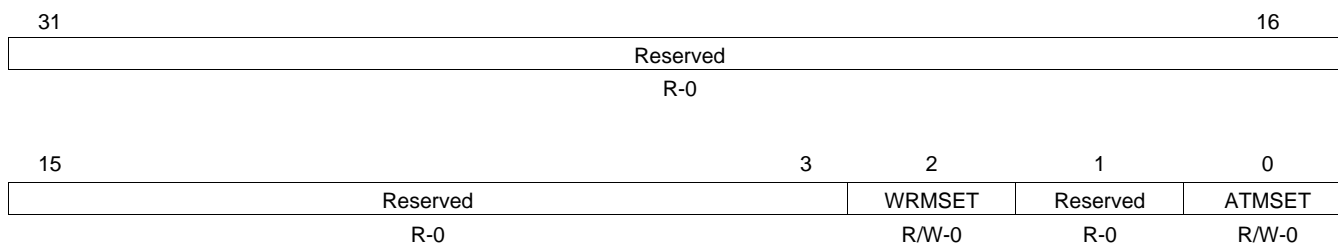
Table 35. EMIF Interrupt Mask Register (EIMR) Field Descriptions

| 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 | WRM | 0 1 | Wait Rise Masked. This bit is set to 1 by hardware to indicate a rising edge has occurred on the EM_WAIT pin, provided that the WRMSET bit is set to 1 in the EMIF interrupt mask set register (EIMSR). 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 EMIF interrupt mask register (EIMR). |
| 1 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0. |
| 0 | ATM | 0 1 | Asynchronous Timeout Masked. This bit is set to 1 by hardware to indicate that during an extended asynchronous memory access cycle the EM_WAIT pin did not go inactive within the number of cycles defined by the MEWC field in the asynchronous wait cycle configuration register (AWCCR), provided that the ATMSET bit is set to 1 in the EMIF interrupt mask set register (EIMSR). 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 and the AT bit in the EMIF interrupt mask register (EIMR). |

4.5 EMIF Interrupt Mask Set Register (EIMSR)

The EMIF interrupt mask set register (EIMSR) is used to enable the interrupts. If a bit is set to 1, the corresponding bit in the EMIF interrupt masked register (EIMR) will be set and an interrupt will be generated when the associated interrupt condition occurs. If a bit is cleared to 0, the the corresponding bit in EIMR will always read 0 and no interrupts will be generated when the associated interrupt condition occurs. Writing a 1 to the WRMSET and ATMSET bits enables each respective interrupt. The EIMSR is shown in Figure 24 and described in Table 36.

Figure 24. EMIF Interrupt Mask Set Register (EIMSR)



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

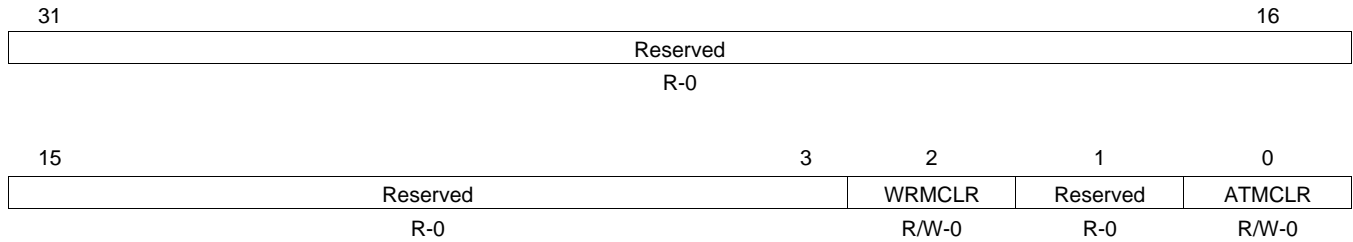
Table 36. EMIF Interrupt Mask Set Register (EIMSR) Field Descriptions

| 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 | WRMSET | 0 1 | 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 WRMCLR bit in the EMIF interrupt mask clear register (EIMCR), and enables the wait rise interrupt. To clear this bit, a 1 must be written to the WRMCLR bit in EIMCR. 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 WRMCLR bit in the EMIF interrupt mask clear register (EIMCR). |
| 1 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0. |
| 0 | ATMSET | 0 1 | 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 ATMCLR bit in the EMIF interrupt mask clear register (EIMCR), and enables the asynchronous timeout interrupt. To clear this bit, a 1 must be written to the ATMCLR bit in EIMCR. 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 ATMCLR bit in the EMIF interrupt mask clear register (EIMCR). |

4.6 EMIF Interrupt Mask Clear Register (EIMCR)

The EMIF interrupt mask clear register (EIMCR) is used to disable the interrupts. If a bit is read as 1, the corresponding bit in the EMIF interrupt masked register (EIMR) will be set and an interrupt will be generated when the associated interrupt condition occurs. If a bit is read as 0, the corresponding bit in EIMR will always read 0 and no interrupt will be generated when the corresponding interrupt condition occurs. Writing a 1 to the ATMCLR and WRMCLR bits disables each respective interrupt. The EIMCR is shown in [Figure 25](#) and described in [Table 37](#).

Figure 25. EMIF Interrupt Mask Clear Register (EIMCR)



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

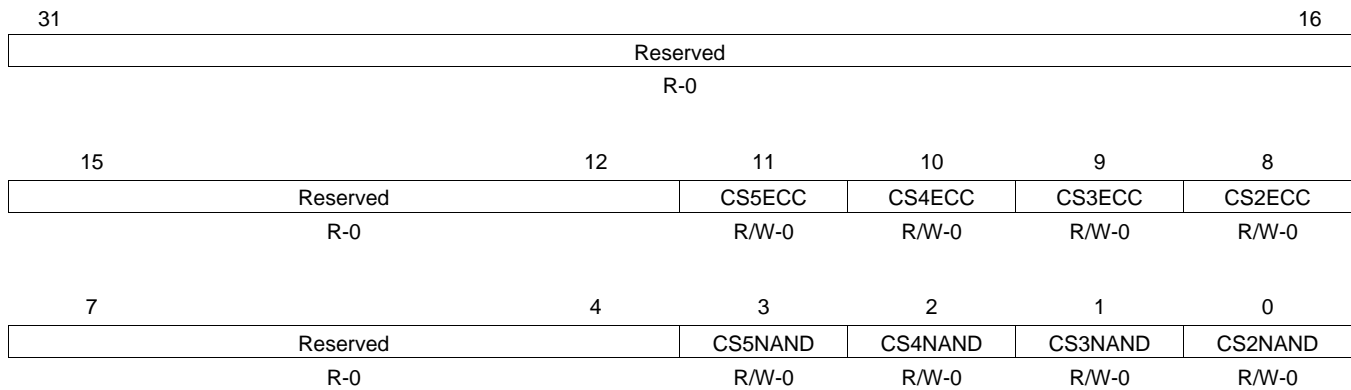
Table 37. EMIF Interrupt Mask Clear Register (EIMCR) Field Descriptions

| 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 | WRMCLR | 0 1 | 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 WRMSET bit in the EMIF interrupt mask set register (EIMSR), and disables the wait rise interrupt. To set this bit, a 1 must be written to the WRMSET bit in EIMSR. 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 WRMSET bit in the EMIF interrupt mask set register (EIMSR). |
| 1 | Reserved | 0 | Reserved. The reserved bit location is always read as 0. If writing to this field, always write the default value of 0. |
| 0 | ATMCLR | 0 1 | 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 ATMSET bit in the EMIF interrupt mask set register (EIMSR), and disables the asynchronous timeout interrupt. To set this bit, a 1 must be written to the ATMSET bit in EIMSR. 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 ATMSET bit in the EMIF interrupt mask set register (EIMSR). |

4.7 NAND Flash Control Register (NANDFCR)

The NAND Flash control register (NANDFCR) is shown in [Figure 26](#) and described in [Table 38](#).

Figure 26. NAND Flash Control Register (NANDFCR)



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

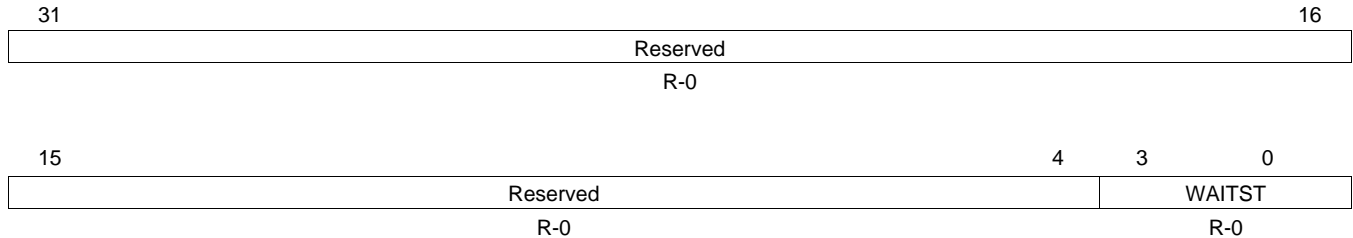
Table 38. NAND Flash Control Register (NANDFCR) Field Descriptions

| Bit | Field | Value | Description |
|-------|----------|-------|---|
| 31-12 | Reserved | 0 | Reserved |
| 11 | CS5ECC | 0 | NAND Flash ECC start for chip select 5. Do not start ECC calculation. |
| | | 1 | Start ECC calculation on data for NAND Flash on $\overline{\text{EM_CS5}}$. |
| 10 | CS4ECC | 0 | NAND Flash ECC start for chip select 4. Do not start ECC calculation. |
| | | 1 | Start ECC calculation on data for NAND Flash on $\overline{\text{EM_CS4}}$. |
| 9 | CS3ECC | 0 | NAND Flash ECC start for chip select 3. Do not start ECC calculation. |
| | | 1 | Start ECC calculation on data for NAND Flash on $\overline{\text{EM_CS3}}$. |
| 8 | CS2ECC | 0 | NAND Flash ECC start for chip select 2. Do not start ECC calculation. |
| | | 1 | Start ECC calculation on data for NAND Flash on $\overline{\text{EM_CS2}}$. |
| 7-4 | Reserved | 0 | Reserved |
| 3 | CS5NAND | 0 | NAND Flash mode for chip select 5. Not using NAND Flash. |
| | | 1 | Using NAND Flash on $\overline{\text{EM_CS5}}$. |
| 2 | CS4NAND | 0 | NAND Flash mode for chip select 4. Not using NAND Flash. |
| | | 1 | Using NAND Flash on $\overline{\text{EM_CS4}}$. |
| 1 | CS3NAND | 0 | NAND Flash mode for chip select 3. Not using NAND Flash. |
| | | 1 | Using NAND Flash on $\overline{\text{EM_CS3}}$. |
| 0 | CS2NAND | 0 | NAND Flash mode for chip select 2. Not using NAND Flash. |
| | | 1 | Using NAND Flash on $\overline{\text{EM_CS2}}$. |

4.8 NAND Flash Status Register (NANDFSR)

The NAND Flash status register (NANDFSR) is shown in [Figure 27](#) and described in [Table 39](#).

Figure 27. NAND Flash Status Register (NANDFSR)



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

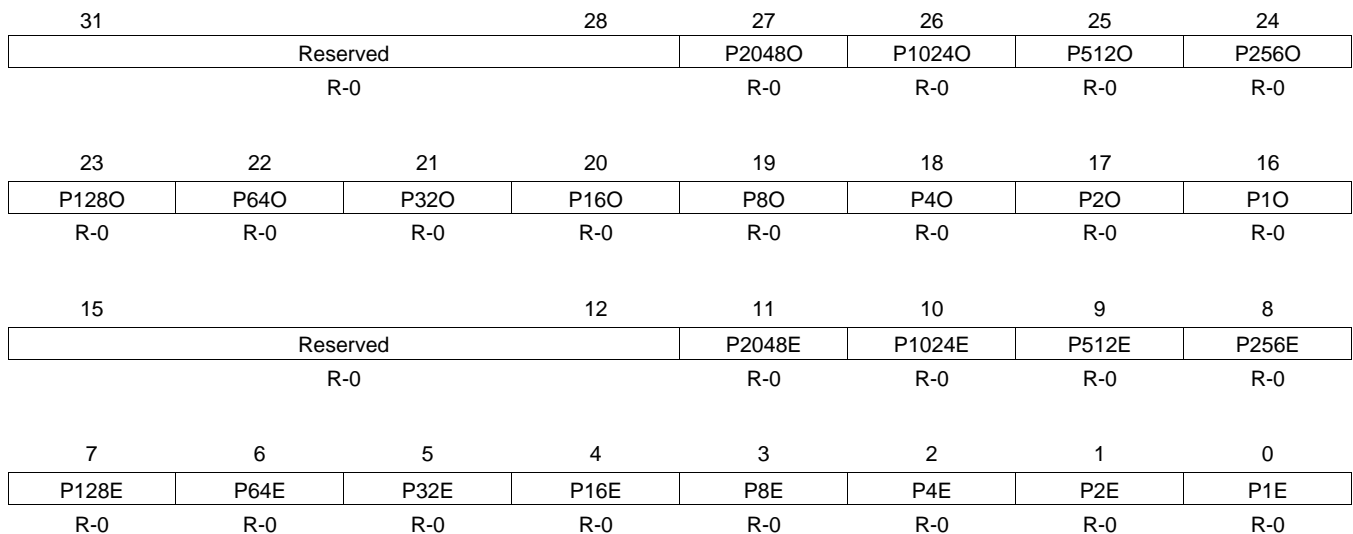
Table 39. NAND Flash Status Register (NANDFSR) Field Descriptions

| Bit | Field | Value | Description |
|------|----------|-------|---|
| 31-4 | Reserved | 0 | Reserved |
| 3-0 | WAITST | 0-Fh | Raw status of the EM_WAIT input pin. The WP0 bit in the asynchronous wait cycle configuration register (AWCCR) has no effect on WAITST. |

4.9 NAND Flash n ECC Registers (NANDF1ECC-NANDF4ECC)

The NAND Flash *n* ECC register (NANDF*n*ECC) is shown in [Figure 28](#) and described in [Table 40](#). For 8-bit NAND Flash, P10, P20, and P40 bits are column parities; P80 to P2048O bits are row parities. For 16-bit NAND Flash, P10, P20, P40, and P80 bits are column parities; P160 to P2048O bits are row parities.

Figure 28. NAND Flash n ECC Register (NANDF*n*ECC)



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

Table 40. 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. |

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

| | |
|-----------------------------|--|
| Amplifiers | amplifier.ti.com |
| Data Converters | dataconverter.ti.com |
| DSP | dsp.ti.com |
| Clocks and Timers | www.ti.com/clocks |
| Interface | interface.ti.com |
| Logic | logic.ti.com |
| Power Mgmt | power.ti.com |
| Microcontrollers | microcontroller.ti.com |
| RFID | www.ti-rfid.com |
| RF/IF and ZigBee® Solutions | www.ti.com/lprf |

Applications

| | |
|--------------------|--|
| Audio | www.ti.com/audio |
| Automotive | www.ti.com/automotive |
| Broadband | www.ti.com/broadband |
| Digital Control | www.ti.com/digitalcontrol |
| Medical | www.ti.com/medical |
| Military | www.ti.com/military |
| Optical Networking | www.ti.com/opticalnetwork |
| Security | www.ti.com/security |
| Telephony | www.ti.com/telephony |
| Video & Imaging | www.ti.com/video |
| Wireless | www.ti.com/wireless |

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2008, Texas Instruments Incorporated