TMS320VC5510 DSP External Memory Interface (EMIF) Reference Guide

Literature Number: SPRU590
August 2004

Texas Instruments

IMPORTANT NOTICE

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

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

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

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

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

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

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

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

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

Preface

Read This First

About This Manual

This manual describes the features and operation of the external memory interface (EMIF) that is available on the TMS320VC5510 digital signal processors (DSPs) in the TMS320C55x™ (C55x™) DSP generation.

Notational Conventions

This document uses the following conventions:

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

  40h

  Similarly, binary numbers often are shown with the suffix b. For example, the following number is the decimal number 4 shown in binary form:

  0100b

- If a signal or pin is active low, it has an overbar. For example, the $\overline{\text{RESET}}$ signal is active low.
- Bits and signals are sometimes referenced with the following notations:

Notation	Description	Example
Register(n–m)	Bits n through m of Register	R(15–0) represents the 16 least significant bits of register R.
Bus[n:m]	Signals n through m of Bus	A[21:1] represents signals 21 through 1 of bus A.

- The following terms are used to name portions of data:

Term	Description	Example
LSB	Least significant bit	In R(15–0), bit 0 is the LSB.
MSB	Most significant bit	In R(15–0), bit 15 is the MSB.
LSW	Least significant word	In R(31–0), bits 15–0 are the LSW.
MSW	Most significant word	In R(31–0), bits 31–16 are the MSW.

Related Documentation From Texas Instruments

The following documents describe the C55x devices and related support tools. 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.

TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual (literature number SPRS076) describes the features of the TMS320VC5510 fixed-point DSP and provides signal descriptions, pinouts, electrical specifications, and timings for the device.

TMS320C55x Technical Overview (literature number SPRU393). This overview is an introduction to the TMS320C55x DSPs, the latest generation of fixed-point DSPs in the TMS320C5000™ DSP platform. Like the previous generations, this processor is optimized for high performance and low-power operation. This book describes the CPU architecture, low-power enhancements, and embedded emulation features.

TMS320C55x DSP CPU Reference Guide (literature number SPRU371) describes the architecture, registers, and operation of the CPU for the TMS320C55x DSPs.

TMS320C55x DSP Peripherals Overview Reference Guide (literature number SPRU317) introduces the peripherals, interfaces, and related hardware that are available on TMS320C55x DSPs.

TMS320C55x DSP Algebraic Instruction Set Reference Guide (literature number SPRU375) describes the TMS320C55x DSP algebraic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the mnemonic instruction set.

TMS320C55x DSP Mnemonic Instruction Set Reference Guide (literature number SPRU374) describes the TMS320C55x DSP mnemonic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the algebraic instruction set.

TMS320C55x Optimizing C/C++ Compiler User's Guide (literature number SPRU281) describes the TMS320C55x™ C/C++ Compiler. This C/C++ compiler accepts ISO standard C and C++ source code and produces assembly language source code for TMS320C55x devices.

TMS320C55x Assembly Language Tools User's Guide (literature number SPRU280) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for TMS320C55x devices.

TMS320C55x DSP Programmer's Guide (literature number SPRU376) describes ways to optimize C and assembly code for the TMS320C55x DSPs and explains how to write code that uses special features and instructions of the DSPs.

Trademarks

TMS320C5000, TMS320C55x, and C55x are trademarks of Texas Instruments.

Other trademarks are the property of their respective owners.

This page is intentionally left blank.

Contents

Figures

Tables

Chapter 1

Introduction to the EMIF

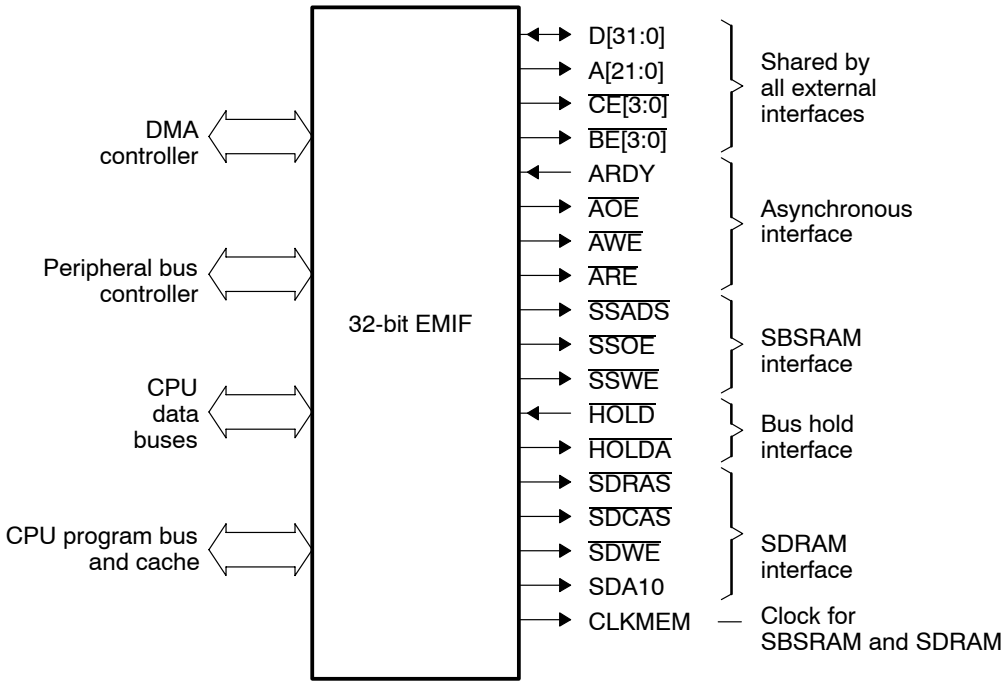
This chapter introduces the external memory interface (EMIF) that is in the TMS320VC5510 DSPs. The EMIF controls all data transfers between the DSP and external memory.

Topic **Page**

1.1 Overview

Figure 1−1 illustrates how the EMIF is interconnected with other parts of the DSP and with external memory devices. The connection to the peripheral bus controller allows the CPU to access the EMIF registers.

Figure 1−1. EMIF Inputs and Outputs



The EMIF provides a glueless interface to three types of memory devices:

- ❏ Asynchronous devices, including ROM, flash memory, and asynchronous SRAM. For details, see Chapter 2.
- ❏ Synchronous burst SRAM (SBSRAM). Details are in Chapter 3.
- ❏ Synchronous DRAM (SDRAM). Details are in Chapter 4.

The EMIF supports the following types of accesses. Details about how data travels on the data bus during each of these types of accesses can be read in Appendix A.

- ❏ 32-bit instruction fetches for the CPU or the instruction cache
- ❏ 32-bit data accesses for the CPU or the DMA controller
- ❏ 16-bit data accesses for the CPU or the DMA controller
- ❏ 8-bit data accesses for the CPU or the DMA controller

To know what parts of the DSP can send requests to the EMIF and the order in which the EMIF services simultaneous requests, see section 1.3, *EMIF Requests*, on page 1-9.

If you want the DSP to share memory chips with an external device, see section 1.6, *HOLD Requests: Sharing External Memory*, on page 1-13.

If you want to buffer CPU write operations to reduce delays, see section 1.7, *Write Posting: Buffering Writes to External Memory*, on page 1-14.

1.2 EMIF Signals

Table 1−1 through Table 1−4 describe the EMIF signals. When the EMIF acknowledges a HOLD request from an external device (see section 1.6 on page 1-13), all output pins except the $\overline{\text{HOLDA}}$ pin are placed in the high-impedance state.

The EMIF data bus has software-controlled bus holders that can be used to hold the bus state when the bus is not being driven. Controlling the bus eliminates additional power consumption due to a floating bus. The EMIF data bus holders are controlled by the BH bit in the system register (SYSR). When BH = 1, the bus holders are enabled on the EMIF data bus. The bus holders are weak drivers that will maintain the last driven state of the bus. The bus holders do not prevent the bus from being driven by the host. When BH = 0, the bus holders are disabled. In multiprocessing structures where multiple DSPs are present with the EMIF data buses connected in parallel, it may be desirable to disable the bus holders on some or all of the DSPs.

For more information on SYSR register functions see the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076).

Table 1–1. EMIF Signals Common Across Memory Types

Signal(s)	Possible States†	Description
$\overline{CE0}$ $\overline{CE1}$ $\overline{CE2}$ $\overline{CE3}$	O/Z	Chip enable pins These active-low pins correspond to the four predefined address ranges (called chip enable (CE) spaces) in the external memory of the DSP. The $\overline{CE}$ pins are meant to be connected to the chip select pins of memory chips. For details about the CE spaces, see section 1.4 on page 1-11.
$\overline{BE[3:0]}$	O/Z	Byte enable pins The EMIF drives signal combinations on these active-low pins to tell the memory chip which pins of the data bus are to be used or ignored during an 8-bit, 16-bit, or 32-bit access. The following diagram shows which byte enable pin corresponds to which byte of the data bus. Byte enable pins: $\overline{BE3}$ $\overline{BE2}$ $\overline{BE1}$ $\overline{BE0}$ Data bus pins: D[31:24] D[23:16] D[15:8] D[7:0] When a byte is written to 16-bit-wide or 32-bit-wide memory, only one byte of the data bus is used; therefore, only one byte enable signal is activated. When a byte is requested from 16-bit-wide or 32-bit-wide memory, the EMIF reads the full width of the memory and internally unpacks the requested byte. As a result, the EMIF activates multiple byte enable signals (two for 16 bits or four for 32 bits). Appendix A shows the behavior of the byte enable signals for various types of accesses.
D[31:0]	I/O/Z	32-bit EMIF data bus The EMIF drives or accepts 32 bits, 16 bits, or 8 bits at a time on these pins. The data pins needed depend on the width of the external memory and the access type; for details, see Appendix A. Between data transfers, the EMIF does not drive the data bus. If the bus holders are enabled, the data bus retains the last driven state. If the bus holders are disabled, the data bus enters the high-impedance state.
A[21:0]	O/Z	22-bit EMIF address bus The EMIF uses this bus to send addresses to memory chips. The external address lines A[21:0] of the EMIF correspond to bits 21–0 of the internal program address. The address pins needed depend on the type of the external memory.

† I = Input; O = Output; Z = High-impedance. The data bus pins and the output pins (except for $\overline{HOLDA}$) are in the high-impedance state while an external device is driving the $\overline{HOLD}$ signal low.

Table 1–1. EMIF Signals Common Across Memory Types (Continued)

Signal(s)	Possible States†	Description
$\overline{\text{HOLD}}$	I	HOLD request pin An external device can drive this active-low signal to request the EMIF to release the bus and control lines connected to EMIF pins. For details about HOLD requests, see section 1.6 on page 1-13.
$\overline{\text{HOLDA}}$	O	HOLD acknowledge pin When the EMIF receives a HOLD request from an external device (on the $\overline{\text{HOLD}}$ pin), the EMIF completes any current activity. Then it places its bus and control pins in the high-impedance state and sends acknowledgement on the $\overline{\text{HOLDA}}$ pin. The external device should wait until $\overline{\text{HOLDA}}$ is driven low before using the bus and control lines. Details about HOLD requests are in section 1.6 (page 1-13).
CLKMEM	O/Z	Memory clock pin (for SBSRAM and SDRAM only) This pin supplies a clock signal for SBSRAM and SDRAM chips. The pin is locked high or driven with the memory clock, depending on the value of the MEMCEN bit. If the memory clock is being driven on the pin (MEMCEN = 1), the frequency of the clock depends on the MEMFREQ bits. MEMCEN and MEMFREQ are in the EMIF global control register, which is described in section 5.2 (page 5-3).

† I = Input; O = Output; Z = High-impedance. The data bus pins and the output pins (except for $\overline{\text{HOLDA}}$) are in the high-impedance state while an external device is driving the $\overline{\text{HOLD}}$ signal low.

Table 1–2. EMIF Signals Specific to Asynchronous Memory

Signal(s)	Possible States†	Description
ARDY	I	Asynchronous ready pin An asynchronous memory chip can drive this active-high signal low whenever it needs to stretch the accesses of the EMIF. For details about using ARDY, see section 2.5 on page 2-10.
$\overline{\text{AOE}}$	O/Z	Asynchronous output enable pin During asynchronous read operations, the EMIF drives $\overline{\text{AOE}}$ low. Connect this active-low pin to the output enable pin of the asynchronous memory chip.

† I = Input; O = Output; Z = High-impedance. $\overline{\text{AOE}}$, $\overline{\text{ARE}}$, and $\overline{\text{AWE}}$ are in the high-impedance state while an external device is driving the $\overline{\text{HOLD}}$ signal low.

Table 1–2. EMIF Signals Specific to Asynchronous Memory (Continued)

Signal(s)	Possible States†	Description
$\overline{\text{ARE}}$	O/Z	Asynchronous read strobe pin The EMIF drives $\overline{\text{ARE}}$ when performing a memory read access. Connect this active-low pin to the read enable pin of the asynchronous memory chip.
$\overline{\text{AWE}}$	O/Z	Asynchronous write strobe pin The EMIF drives $\overline{\text{AWE}}$ when performing a memory write access. Connect this active-low pin to the write enable pin of the asynchronous memory chip.

† I = Input; O = Output; Z = High-impedance. $\overline{\text{AOE}}$, $\overline{\text{ARE}}$, and $\overline{\text{AWE}}$ are in the high-impedance state while an external device is driving the $\overline{\text{HOLD}}$ signal low.

Table 1–3. EMIF Signals Specific to SBSRAM

Signal(s)	Possible States†	Description
$\overline{\text{SSADS}}$	O/Z	Address strobe/enable pin for SBSRAM The EMIF drives $\overline{\text{SSADS}}$ is low (active) while it drives addresses on the address bus.
$\overline{\text{SSOE}}$	O/Z	Output buffer enable pin for SBSRAM Connect this active-low pin to the output enable pin of the SBSRAM chip. The EMIF drives this pin high or low to disable or enable the data output drivers of the SBSRAM chip.
$\overline{\text{SSWE}}$	O/Z	Write enable pin for SBSRAM Connect this active-low pin to the write enable pin of the SBSRAM chip. The EMIF drives this pin low while it writes data to the SBSRAM chip.

† I = Input; O = Output; Z = High-impedance. The SBSRAM pins are in the high-impedance state while an external device is driving the $\overline{\text{HOLD}}$ signal low.

Table 1–4. EMIF Signals Specific to SDRAM

Signal(s)	Possible States†	Description
$\overline{\text{SDRAS}}$	O/Z	Row strobe pin for SDRAM $\overline{\text{SDRAS}}$ is active (low) during the ACTV, DCAB, REFR, and MRS commands.
$\overline{\text{SDCAS}}$	O/Z	Column strobe pin for SDRAM $\overline{\text{SDCAS}}$ is active (low) during reads and writes, and during the REFR and MRS commands.
$\overline{\text{SDWE}}$	O/Z	Write enable pin for SDRAM $\overline{\text{SDWE}}$ is active (low) during writes and during the DCAB and MRS commands.
SDA10	O/Z	A10 address line for SDRAM This signal acts as a row address bit during ACTV commands. SDA10 also acts as the auto-precharge enable for SDRAM memories during read and write operations. SDA10 is active (high) during a DCAB command.

† I = Input; O = Output; Z = High-impedance. The SDRAM pins are in the high-impedance state while an external device is driving the $\overline{\text{HOLD}}$ signal low.

1.3 EMIF Requests

The EMIF services the requests shown in Table 1–5. If multiple requests arrive simultaneously, the EMIF prioritizes them as shown in the Priority column.

Table 1–5. EMIF Requests and Their Priorities

EMIF Requester	Priority	Description
HOLD	1 (highest)	A request initiated when an external device drives the $\overline{\text{HOLD}}$ pin low. For details on HOLD requests, see section 1.6 on page 1-13.
Urgent refresh	2	A request from synchronous DRAM that needs an immediate refresh
E bus	3	A write request from the E bus of the DSP CPU
F bus	4	A write request from the F bus of the DSP CPU
D bus	5	A read request from the D bus of the DSP CPU
C bus	6	A read request from the C bus of the DSP CPU
P bus	7	An instruction fetch request from the DSP CPU or instruction cache. In the CPU, instructions are received on the P bus.
DMA controller	8	A write or read request from the DSP DMA controller
Trickle refresh	9 (lowest)	A request from synchronous DRAM that needs the next periodic refresh

As shown in Table 1–6, there is a subtle difference between dual data accesses and long data accesses requested by the CPU. The following two instructions are examples of these access types:

```
ADD *AR0, *AR1, AC0    ; Dual data access. Two separate
                       ; 16-bit values referenced by
                       ; pointers AR0 and AR1.

ADD dbl(*AR2), AC1     : Long data access. One 32-bit
                       ; value referenced by pointer AR2.
```

Both access types require two 16-bit data buses in the CPU, but they require different numbers of EMIF requests. A dual data access involves two separate 16-bit values and, therefore, requires two EMIF requests. A long data access involves a single 32-bit value and, therefore, a single EMIF request. This EMIF request corresponds to the address bus used. For example, if a long data read is performed, the DAB address bus is used, and the EMIF receives a D-bus request.

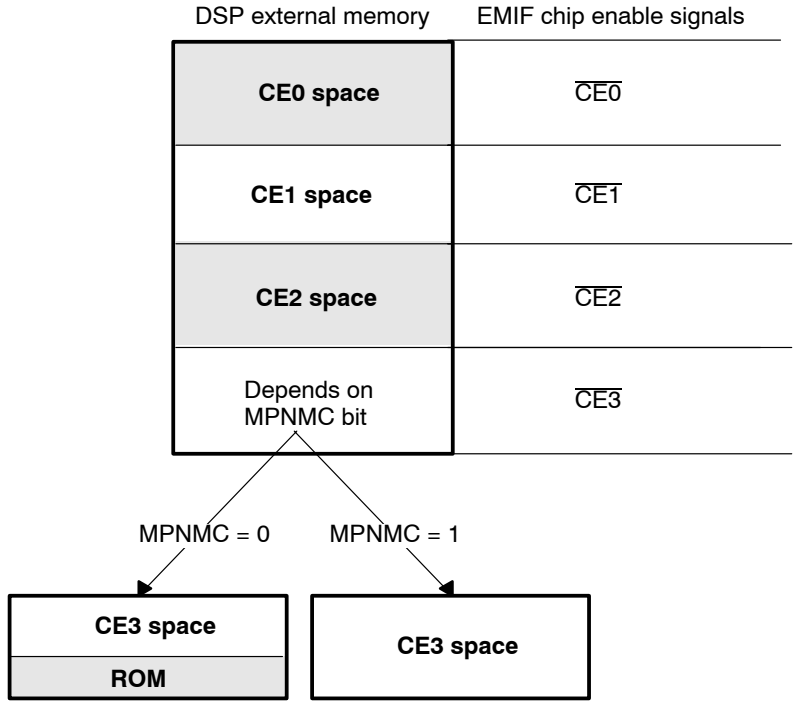
Table 1–6. EMIF Requests Associated With Dual and Long Data Accesses

Access Type	CPU Data Buses Used	CPU Address Bus(es) Used	Request(s) Sent To EMIF
Dual data read	CB and DB (carrying two 16-bit values)	CAB and DAB	C-bus request to read 16 bits D-bus request to read 16 bits
Dual data write	EB and FB (carrying two 16-bit values)	EAB and FAB	E-bus request to write 16 bits F-bus request to write 16 bits
Long data read	CB and DB (carrying one 32-bit value)	DAB	D-bus request to read 32 bits
Long data write	EB and FB (carrying one 32-bit value)	EAB	E-bus request to write 32 bits

1.4 CE Spaces in the External Memory Map

The external memory map (see Figure 1–2) is divided into four ranges called chip enable (CE) spaces. The CE0 space is at the lowest addresses, and the CE3 space is at the highest addresses. Each CE space can contain up to 4M bytes. Some of the addresses in the CE3 space can be used to access ROM inside the DSP. As shown in Figure 1–2, you can switch these addresses between the CE3 space and the ROM by changing the MPNMC bit in CPU status register ST3_55. MPNMC and ST3_55 are described in the *TMS320C55x DSP CPU Reference Guide* (SPRU371). For the number of bytes in each CE space and in the ROM, see the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076).

Figure 1–2. CE Spaces and the Associated $\overline{CE}$ Signals



1.4.1 Chip Enable ($\overline{CE}$) Pins for the CE Spaces

Each CE space has an associated chip enable ($\overline{CE}$) pin on the EMIF (see the right side of Figure 1–2). To assign an asynchronous memory chip or an SBSRAM chip to one of the CE spaces, connect the associated $\overline{CE}$ pin to the chip select pin of the chip. When the EMIF performs an access to a CE space, it drives the associated $\overline{CE}$ pin low. For example, you can assign an SBSRAM memory chip to the CE1 space by connecting $\overline{CE1}$. When the EMIF performs an access in the CE1 space, it drives $\overline{CE1}$ low, selecting the SBSRAM chip.

However, for SDRAM chips, the $\overline{CE}$ pins are designed to operate differently. The SDRAM chip arrangements supported by the EMIF fill two or four of the CE spaces. However, in most cases, you will connect only one $\overline{CE}$ pin to the chip or chips in the arrangement. For more details, see the examples in section 4.3 (page 4-7).

1.4.2 Defining the Memory Type for Each CE Space

In addition to connecting chip enable pins, you must indicate which type of memory is in each CE space. For each CE space, you must write the appropriate value to the MTYPE field of CE space control register 1. Table 1–7 shows the options. For a description of the CE space registers, see section 5.5 on page 5-11.

Table 1–7. Specifying a Memory Type With the MTYPE Bits

MTYPE	Memory Type
000b†	Asynchronous, 8-bit data bus width
001b	Asynchronous, 16-bit data bus width
010b	Asynchronous, 32-bit data bus width
011b	SDRAM, 16-bit or 32-bit data bus width
100b	SBSRAM, 32-bit data bus width
other	Reserved

† When MTYPE = 000b, the EMIF can only read from external memory; writes to 8-bit-wide memory are not possible.

1.5 Latency For a Change of CE Space or Data Direction

If there is a change of CE space or data direction (read/write) between two consecutive EMIF requests, the EMIF inserts one dead cycle. During this dead cycle all chip enable signals are driven high (inactive). During accesses to asynchronous memory, this cycle is the cycle that is automatically added to the extended hold period.

1.6 HOLD Requests: Sharing External Memory

The EMIF provides the following two signals to allow an external device and the EMIF to share the external bus and control lines:

- $\overline{\text{HOLD}}$ (HOLD request pin): The external device drives this active-low pin to request exclusive access to the external bus and control lines. The $\overline{\text{HOLD}}$ signal is synchronized as it enters the EMIF. A HOLD request is the highest priority request that the EMIF can receive during active operation. When it receives the HOLD request, the EMIF stops accessing the external memory at the earliest possible moment, which may entail completion of the current accesses, required SDRAM refreshes, and memory device deactivation.
- $\overline{\text{HOLDA}}$ (HOLD acknowledge pin): Once $\overline{\text{HOLD}}$ has been asserted, the EMIF places all of its other output pins in the high-impedance state and then asserts $\overline{\text{HOLDA}}$ by driving it low. The external device may then use the pins of the external memory without the chance of interference from the EMIF.

The external device must continue to drive $\overline{\text{HOLD}}$ low for as long as it wants to use the bus. No SDRAM refreshes occur while $\overline{\text{HOLD}}$ is kept low; however, when $\overline{\text{HOLD}}$ is driven high again, any required SDRAM refreshes occur before pending access requests are serviced.

You can block HOLD requests for all CE spaces by setting the NOHOLD bit (see page 5-5), which is a bit of the EMIF global control register.

While the external device has control of the memory, the CPU can continue to execute instructions (from internal memory) or stop. You choose one of these options by writing to the HOLD mode bit (HM) of status register ST1_55. ST1_55 is a register in the CPU. HM and ST1_55 are described in the *TMS320C55x DSP CPU Reference Guide* (SPRU371).

1.7 Write Posting: Buffering Writes to External Memory

Ordinarily, when a CPU request arrives at the EMIF, the EMIF does not send acknowledgment to the CPU until the EMIF has driven the data on the external bus. As a result, the CPU does begin the next operation until the data is actually sent to the external memory.

If write posting is enabled, the EMIF acknowledges the CPU as soon as the EMIF receives the address and data. The address and data are stored in dedicated write posting registers in the EMIF. When a time slot becomes available, the EMIF runs the posted write operation. If the next CPU access is not for the EMIF and is for internal memory, that access is able to run concurrently with the posted write operation.

The EMIF supports two levels of write posting. That is, the write posting registers can hold data and addresses for up to two CPU accesses at a time. The EMIF allocates the write posting registers on a first requested, first served basis. However, if the E bus and the F bus make requests simultaneously, the E bus is given priority.

To enable write posting for all CE spaces, set the WPE bit in the EMIF global control register. WPE is described on page 5-4. It might be useful to disable write posting (WPE = 0) during debugging.

There are no write posting registers for requests from the DMA controller. However, the EMIF sends acknowledgement to the DMA controller prior to the actual write to external memory. This early acknowledgement allows the DMA controller to transfer the next address early, to avoid dead cycles during burst transfers or between back-to-back single transfers.

1.8 CPU Instruction Pipeline Considerations

This section explains two special cases of pipeline operation that could impact external memory accesses.

As described in the *TMS320C55x DSP CPU Reference Guide* (SPRU371), the CPU uses instruction pipelining. Multiple instructions are processed simultaneously in the pipeline, and different instructions may access external memory during different phases of completion. The pipeline consists of a number of phases during which different, designated tasks are performed.

1.8.1 A Write Followed by a Read at a Different Address

The *read phase* of the pipeline is used to read operands and other data needed to complete the execution of an instruction. The results of an instruction may be written to external memory in the *write phase* of the pipeline. The read phase occurs earlier in the pipeline than the write phase. For this reason, the read and write requests made to the EMIF may occur in an order which is different than the order in which the instructions entered the pipeline. For example, consider the following code segment:

```
I1:    MOV T0, *(#External_Address_1)     ; Instruction 1 writes to external memory
I2:    MOV *(#External_Address_2), T1     ; Instruction 2 reads from external memory
```

Although the code shows the write followed by the read, the read actually occurs first on the EMIF because of the pipelining effect, as shown in Figure 1–3. The figure shows the two instructions passing through the read (R), execute (X), and write (W) phases.

For some applications, maintaining the proper write-read order is critical. In such cases, NOP (no operation) instructions (or other instructions that do not perform external memory accesses) can be inserted between the original instructions to delay the read operation. This technique is shown in Figure 1–4.

Figure 1–3. Partial Pipeline Diagram of Consecutive Instructions That Write and Read at Different Addresses

R	X	W	Cycle	Comment
I1			n	
I2	I1		n+1	Read initiated by instruction 2.
	I2	I1	n+2	Write initiated by instruction 1.
		I2	n+3	

Figure 1–4. NOP Instructions Inserted in the Code of Figure 1–3 to Make the Write Occur Before the Read

R	X	W	Cycle	Comment
I1			n	
NOP	I1		n+1	
NOP	NOP	I1	n+2	Write initiated by instruction 1.
I2	NOP	NOP	n+3	Read initiated by instruction 2.
	I2	NOP	n+4	
		I2	n+5	

1.8.2 A Write Followed by a Read at the Same Address

In most cases, when a write to memory is followed immediately by a read at the same address, the data written is the same data expected back during the read. The C55x CPU takes advantage of this fact with a special memory-bypass feature. During the read, the CPU gets a copy of the data directly from the write bus(es) instead of accessing memory.

When the CPU is accessing external devices, there may be cases in which the memory-bypass feature would lead to unwanted results. For example, suppose two physical memory locations X and Y are mapped to the same address. Writing modifies location X, and reading gets data from location Y. If the memory-bypass feature takes effect, location Y is not read.

To prevent the memory bypass, insert three or more NOP instructions (or other instructions) between the instructions that perform the write and the read. For example:

```
MOVT0,*(#External_Address_1)    ; Write to address 1
NOP                             ; 3-cycle delay
NOP
NOP
MOV*(#External_Address_1), T1   ; Read from address 1
```

1.9 Power and Reset Considerations

1.9.1 Power Conservation

The DSP is divided into idle domains that can be programmed to be idle or active. The state of all domains is called the idle configuration. Any idle configuration that disables the clock generator domain and/or the EMIF domain stops activity in the EMIF. If you program the EMIF domain to be idle, the EMIF completes all data transfers before entering the idle state. While in the idle state, the EMIF does not perform SDRAM refreshes or handle HOLD requests.

1.9.2 Effects of a DSP Reset

If you drive the $\overline{\text{RESET}}$ signal low, the DSP undergoes a reset. A DSP reset resets the EMIF and the EMIF registers. The register figures in Chapter 5 indicate the effects on the register contents.

The EMIF is forced into the HOLD state during reset; therefore, the $\overline{\text{HOLDA}}$ signal is low and all other output signals are in the high-impedance state. After reset, if the $\overline{\text{HOLD}}$ signal is not active, the EMIF leaves the HOLD state and $\overline{\text{HOLDA}}$ goes high.

This page is intentionally left blank.

Chapter 2

Using Asynchronous Memory

The EMIF offers configurable timing parameters so that you can interface the DSP to a variety of asynchronous memory types, including flash memory, SRAM, and EPROM. This chapter describes the EMIF features that support accesses to asynchronous memory.

Topic **Page**

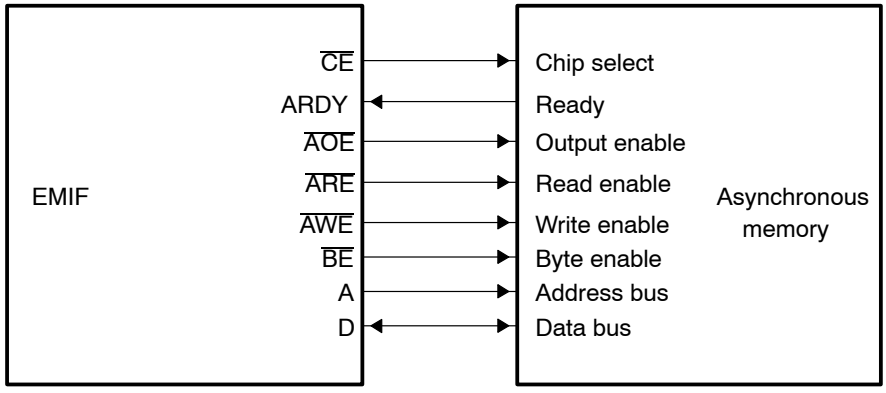
2.1 Interfacing to External Asynchronous Memory

Figure 2−1 shows generalized connections between the EMIF and an asynchronous memory chip. The address pins use depend on the width of the memory:

Memory Width	Address Pins
8 bits	A[21:0]
16 bits	A[21:1]
32 bits	A[21:2]

For other details about the EMIF pins shown in the figure, see section 1.2 on page 1-4.

Figure 2−1. EMIF Connected to an Asynchronous Memory Chip



2.2 Configuring the EMIF for Asynchronous Accesses

To configure the EMIF for interfacing to asynchronous memory, program the following register fields. The EMIF register descriptions are in Chapter 5.

- Each CE space has control registers 1, 2, and 3. For every CE space that contains asynchronous memory:
  - Use the MTYPE field in CE space control register 1 to select one of the options in the following table. When MTYPE = 000b, the EMIF can only read from external memory; writes to 8-bit-wide memory are not possible.

MTYPE	Memory Type
000b	8-bit-wide asynchronous
001b	16-bit-wide asynchronous
010b	32-bit-wide asynchronous

  - Use the other fields in the CE space control registers to program the timing parameters described in Table 2–1. These parameters are ignored unless the MTYPE field indicates asynchronous memory.
- The EMIF global control register has two fields that affect all CE spaces, regardless of memory type:
  - WPE. This bit enables or disables write posting (see section 1.7 on page 1-14) for all CE spaces.
  - NOHOLD. This bit enables or disables HOLD requests (see section 1.6 on page 1-13) for all CE spaces.
- The EMIF global control register also contains the ARDYOFF bit, which is used to enable or disable the ARDY pin. The function of the ARDY pin is described in section 2.5 (page 2-10).

Table 2–1. Parameters for an Access of External Asynchronous Memory

Parameters	Control Bits	Definition
Setup periods	RDSETUP WRSETUP	A setup period is the time in CPU clock cycles given to setup the address, chip enable ($\overline{CE}$), and byte enable ($\overline{BE}$) signals before the read strobe signal ($\overline{ARE}$) or write strobe signal ($\overline{AWE}$) falls. For an asynchronous read operation, this is also the setup period for the output enable signal ($\overline{AOE}$) before $\overline{ARE}$ falls.
Strobe periods	RDSTROBE WRSTROBE	A strobe period is the time in CPU clock cycles between the falling (activation) and rising (deactivation) of the read or write strobe signal.

Table 2–1. Parameters for an Access of External Asynchronous Memory (Continued)

Parameters	Control Bits	Definition
Hold periods	RDHOLD WRHOLD	A hold period is the time in CPU clock cycles during which the address and byte enable lines are held active after the read or write strobe signal rises. For an asynchronous read operation, this is also the hold period for the output enable signal after $\overline{\text{ARE}}$ rises.
Extended hold periods	RDEXHLD WREXHLD	An extended hold period is the number of additional CPU cycles inserted when (a) the EMIF must switch to a different CE space before performing the next access, or (b) the next access requires a change in the data direction (for example, the EMIF has completed a read access and must now perform a write access). All chip enable signals are inactive during this period. The EMIF automatically adds 1 cycle to any cycles you have programmed. For example, if WREXHLD = 0, the extended hold period is 1 cycle.
Time-out value	TIMOUT	A single time-out value applies to both read operations and write operations. During an operation, an internal counter counts the number of CPU clock cycles that the ARDY signal is sampled low (indicating that the memory is not ready for an access). If the counter reaches the time-out value, the EMIF records an error in the bus error status register (see section 5.4 on page 5-7). If a CPU bus requested the memory access, the EMIF sends a bus-error interrupt request to the CPU. If the DMA controller requested the memory access, the EMIF sends a time-out signal to the DMA controller. The DMA controller can ignore the signal or send a bus-error interrupt request to the CPU. The bus error interrupt is maskable; the CPU ignores it or services it depending on whether the interrupt is properly enabled. **Note:** If ARDYOFF = 1 in the EMIF global control register, ARDY is not sampled, and as a result, a time-out condition cannot occur.

Table 2–2. Setup Period Cycles For First and Subsequent Accesses

SETUP Bits	Setup Period For First Access (CPU Clock Cycles)	Setup Period For Subsequent Accesses (CPU Clock Cycles)
0	2	1
1	2	1
2	2	1
$3 \leq n \leq 15$	n	n

2.3 Asynchronous Read Operations

Table 2–3 describes the signal activity of the EMIF when it reads from an asynchronous memory chip. Figure 2–2 is an example timing diagram. The write strobe signal ($\overline{\text{AWE}}$) is driven high (inactive) during a read operation. For more detailed timing information, see the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076).

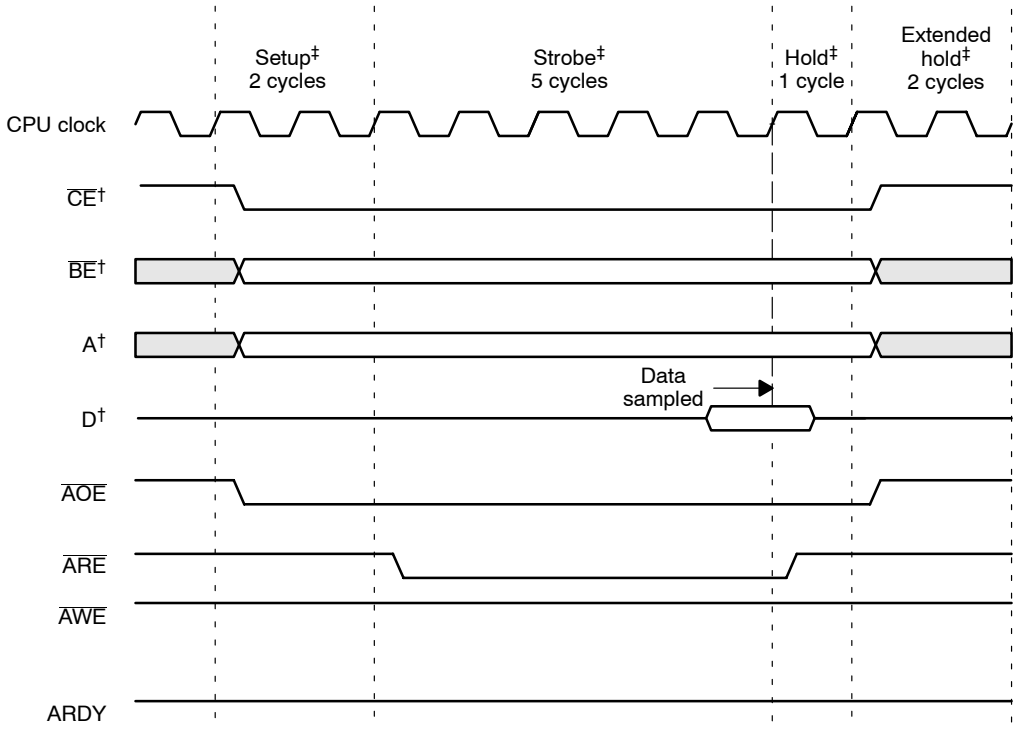
Table 2–3. EMIF Signal Activity During an Asynchronous Read Operation

Time Interval	Signal Activity
Start of setup period	**If this is the first memory read cycle in the selected CE space,** the following signal changes occur simultaneously (n is the number of the selected CE space): ❑ $\overline{\text{CE}}$n falls to enable the memory chip. ❑ The byte enable lines become valid to indicate the number and positions of the bytes being accessed. ❑ The address lines become valid to carry the address to the memory chip. ❑ $\overline{\text{AOE}}$ falls to activate the output enable function of the memory chip. **If this is a subsequent read cycle,** $\overline{\text{CE}}$n and $\overline{\text{AOE}}$ stay low, while new values become valid on the byte enable and address lines. **Note:** In any asynchronous read operation, the setup period is a minimum of 2 cycles for the first memory access. After the first access, the setup period is a minimum of 1 cycle.
Strobe period (may be extended with ARDY)	1) $\overline{\text{ARE}}$ falls to mark the start of the strobe period. 2) In Figure 2–2, ARDY is inactive. However, if the programmed strobe period is 4 cycles or greater, the memory chip can drive ARDY low to request additional cycles if it needs more time to provide the data. For more details on using ARDY, see section 2.5 on page 2-10. 3) Data is sampled by the EMIF on the CPU clock rising edge that is concurrent with the end of the strobe period, just prior to the rising of $\overline{\text{ARE}}$. 4) $\overline{\text{ARE}}$ rises.

Table 2–3. EMIF Signal Activity During an Asynchronous Read Operation (Continued)

Time Interval	Signal Activity
End of hold period	If there is another read access in the same CE space, new values become valid on the byte enable and address lines to begin a new memory cycle. Otherwise, the following signal changes occur simultaneously: ❑ The byte enable lines become invalid. ❑ The address lines become invalid. ❑ $\overline{\text{AOE}}$ rises to deactivate the output enable function of the memory chip.
Extended hold period	All chip enable ($\overline{\text{CE}}$) signals are deactivated, to prevent accidental contention while the active memory chip is disabled and another memory chip is enabled. The extended hold period is only inserted if the EMIF receives no new access request in the same CE space and with the same data direction (read, in this case). If a request of the same type occurs during this period, the extended hold is aborted and a new memory cycle begins.

Figure 2–2. Asynchronous Read Operation



† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.
‡ These periods are programmable. The values shown here are for example only.

2.4 Asynchronous Write Operations

Table 2–4 explains signal activity during a write to asynchronous memory. Figure 2–3 is an example timing diagram. The output enable signal ($\overline{\text{AOE}}$) and the read strobe signal ($\overline{\text{ARE}}$) are driven high (inactive) during a write operation. For more detailed timing information, see the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076).

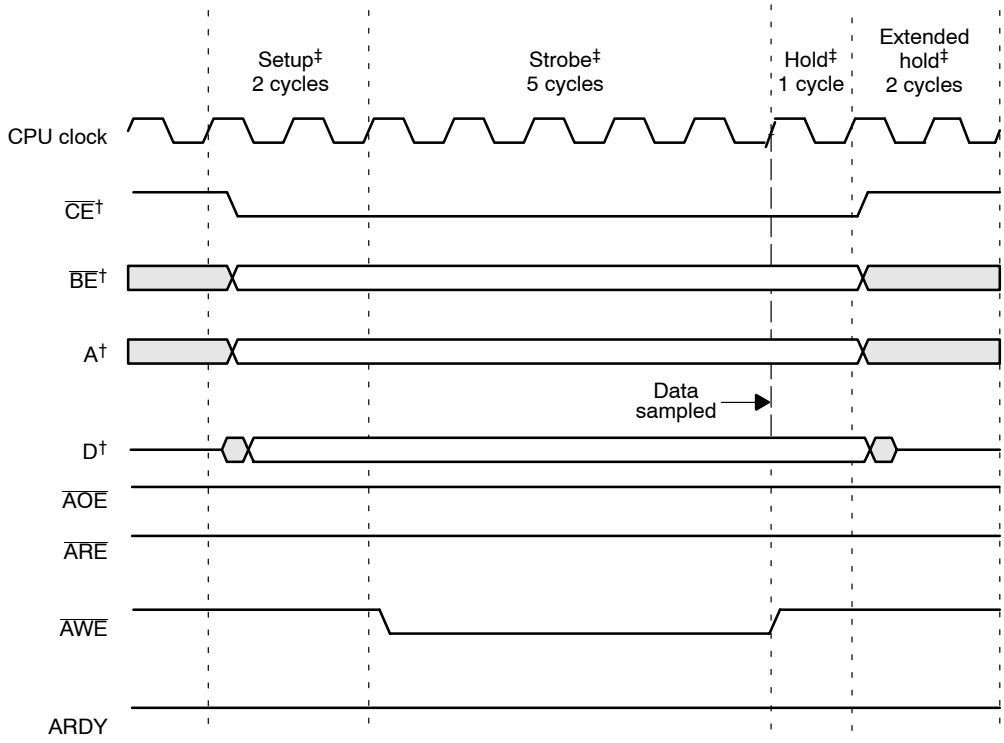
Table 2–4. EMIF Signal Activity During an Asynchronous Write Operation

Time Interval	Signal Activity
Start of setup period	**If this is first memory write cycle in the selected CE space,** the following signal changes occur (n is the number of the selected CE space): ❑ $\overline{\text{CE}}$n falls to enable the memory chip. ❑ The byte enable lines become valid to indicate the number and positions of the bytes being accessed. ❑ The address lines become valid to carry the address to the memory chip. ❑ The data lines become valid to carry data to the memory chip. **If this is a subsequent write cycle,** $\overline{\text{CE}}$n stays low, while new values become valid on the byte enable, address, and data lines. **Note:** In any asynchronous write operation, the setup period is a minimum of 2 cycles for the first memory access. After the first access, the setup period is a minimum of 1 cycle.
Strobe period (may be extended with ARDY)	1) $\overline{\text{AWE}}$ falls to mark the start of the strobe period. 2) In Figure 2–3, ARDY is inactive. However, if the programmed strobe period is 4 cycles or greater, the memory chip can drive ARDY low to request additional cycles if it needs more time to accept the data. More details on using ARDY are in section 2.5 (page 2-10). 3) The external memory chip latches the data on the rising edge of $\overline{\text{AWE}}$.

Table 2–4. EMIF Signal Activity During an Asynchronous Write Operation (Continued)

Time Interval	Signal Activity
End of hold period	If there is another write access in the same CE space, new values become valid on the byte enable, address, and data lines to begin a new memory cycle. Otherwise, the following signal changes occur: ❑ The byte enable lines become invalid. ❑ The address lines become invalid. ❑ The data lines become invalid.
Extended hold period	All chip enable ($\overline{CE}$) signals are deactivated, to prevent accidental contention while the active memory chip is disabled and another memory chip is enabled. The extended hold period is only inserted if the EMIF receives no new access request in the same CE space and with the same data direction (write, in this case). If a request of the same type occurs during this period, the extended hold is aborted and a new memory cycle begins.

Figure 2–3. Asynchronous Write Operation



† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.
‡ These periods are programmable. The values shown here are for example only.

2.5 Inserting Extra Cycles With the Ready (ARDY) Signal

In addition to programmable access shaping, you can insert extra cycles into the strobe period by activating the ARDY signal. This ready input signal is internally synchronized to the CPU clock.

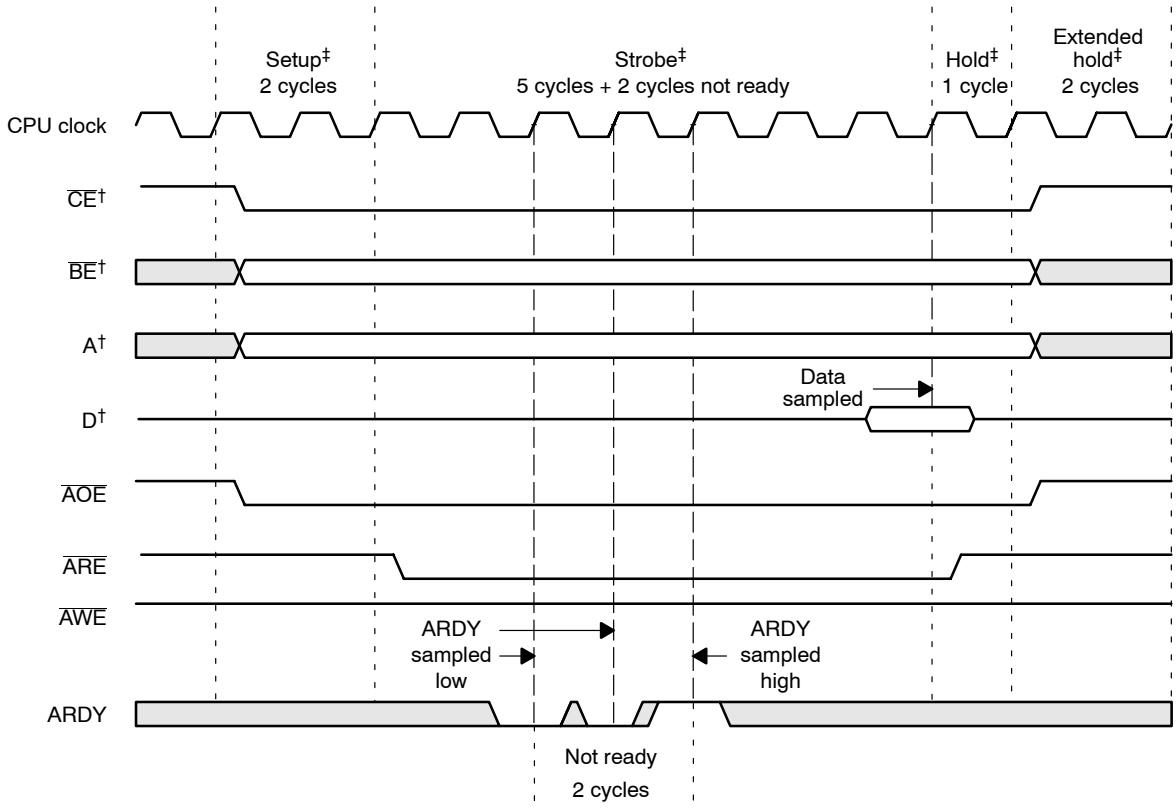
The EMIF checks ARDY on the third rising edge of the CPU clock before the end of the programmed strobe period. If ARDY is detected low at that time, the strobe period is extended by 1 CPU clock cycle. For each subsequent CPU clock rising edge that ARDY is sampled low, the strobe period is extended by 1 CPU clock cycle. Thus, if ARDY is to be used, the minimum strobe period must be 4. If the strobe period is less than 4, ARDY must be held high; otherwise, unexpected behavior may occur in the EMIF. The maximum number of cycles that ARDY can be used depends on the time-out value in the TIMOUT bits. A time-out counter in the EMIF starts to count on the third rising edge of the CPU clock cycle before the end of the programmed strobe period.

If ARDY extends the strobe period, the EMIF does not allow a hold period of 0 cycles. If programmed hold period is 0 cycles and ARDY is used, the EMIF automatically generates a 1-cycle hold period.

Figure 2−4 shows one case where ARDY is used to extend the strobe period during a read operation. Figure 2−5 shows a similar case for a write operation.

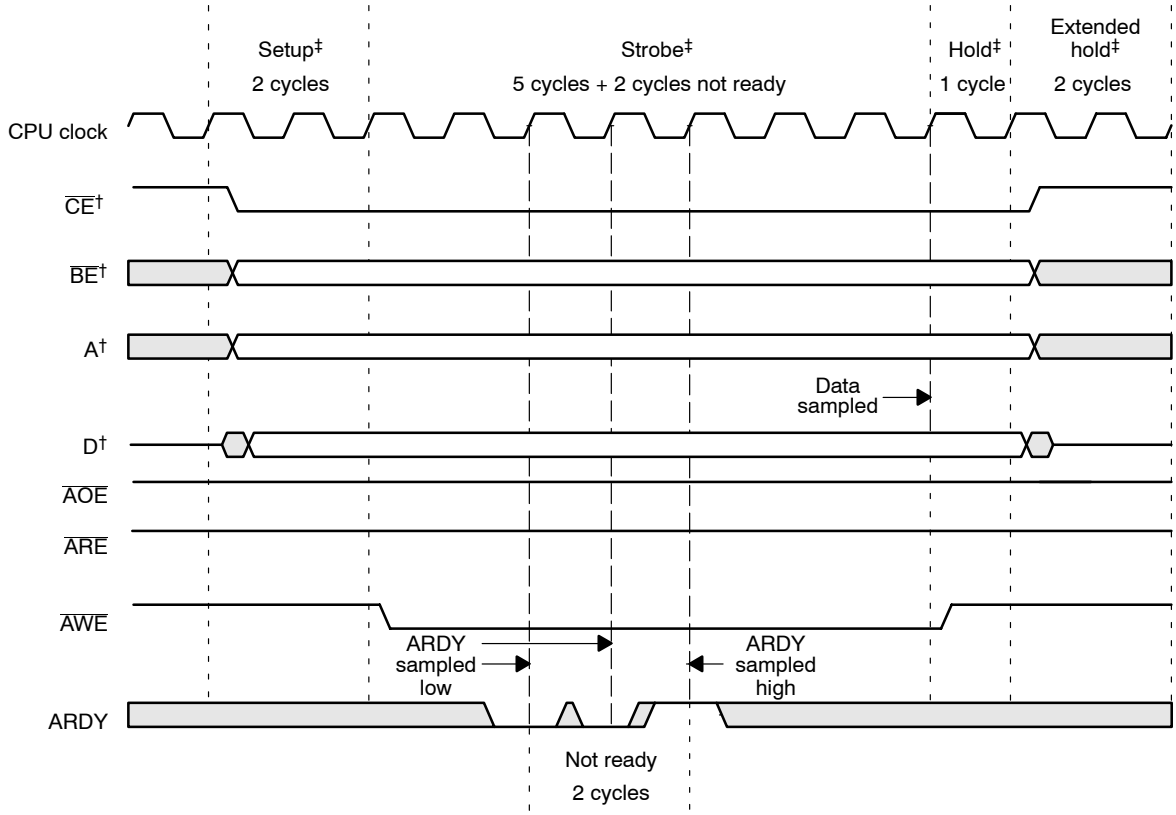
If you want the EMIF to ignore the ARDY signal, set the ARDYOFF bit in the EMIF global control register. When ARDYOFF = 1, the ARDY signal is not sampled, and, as a result, the strobe period is not extended beyond the programmed length.

Figure 2–4. Extending an Asynchronous Read Operation With the ARDY Signal



† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.
‡ These periods are programmable. The values shown here are for example only.

Figure 2–5. Extending an Asynchronous Write Operation With the ARDY Signal



† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.
‡ These periods are programmable. The values shown here are for example only.

Chapter 3

Using SBSRAM

This chapter describes how to program the EMIF to read from and write to 32-bit-wide industry standard synchronous burst SRAM (SBSRAM) chip. SBSRAM chips are available in both flow through and pipeline types; however, this EMIF supports only pipeline SBSRAM, which has the capability to operate at higher frequencies with sustained throughput. The SBSRAM interface can be programmed to run at the frequency of the CPU clock or at half the frequency of the CPU clock; however, the frequencies you can choose depend on the timing limitations shown in the electrical specifications section of the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076).

Topic **Page**

3.1 Configuring the EMIF for SBSRAM Accesses

To configure the EMIF for interfacing to SBSRAM, program the following register fields. The EMIF register descriptions are in Chapter 5.

- Each CE space has a CE space control register 1 that includes a memory type (MTYPE) field. For each CE space that contains SBSRAM, define the memory type as SBSRAM (MTYPE = 100b).
- The EMIF global control register has two fields for controlling the memory clock signal, which is used for all SBSRAM and all SDRAM:
  - MEMCEN. This bit enables or disables the memory clock on the CLKMEM pin.
  - MEMFREQ. When the memory clock is disabled (MEMCEN = 0), write to MEMFREQ to select the relative frequency of the memory clock signal. Your options are 1 or 1/2 times the frequency of the CPU clock signal. The frequencies you can choose depend on the timing limitations shown in the electrical specifications section of the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076). After programming MEMFREQ, enable the clock (MEMCEN = 1).
- The EMIF global control register also has two fields that affect all CE spaces, regardless of memory type:
  - WPE. This bit enables or disables write posting (see section 1.7 on page 1-14) for all CE spaces.
  - NOHOLD. This bit enables or disables HOLD requests (see section 1.6 on page 1-13) for all CE spaces.

3.2 Examples of Interfacing to SBSRAM Chips

The following examples are provided to help you to connect pins and to configure the EMIF for various SBSRAM chips. For details about the EMIF pins shown in the examples, see section 1.2 on page 1-4. Following are some interfacing considerations:

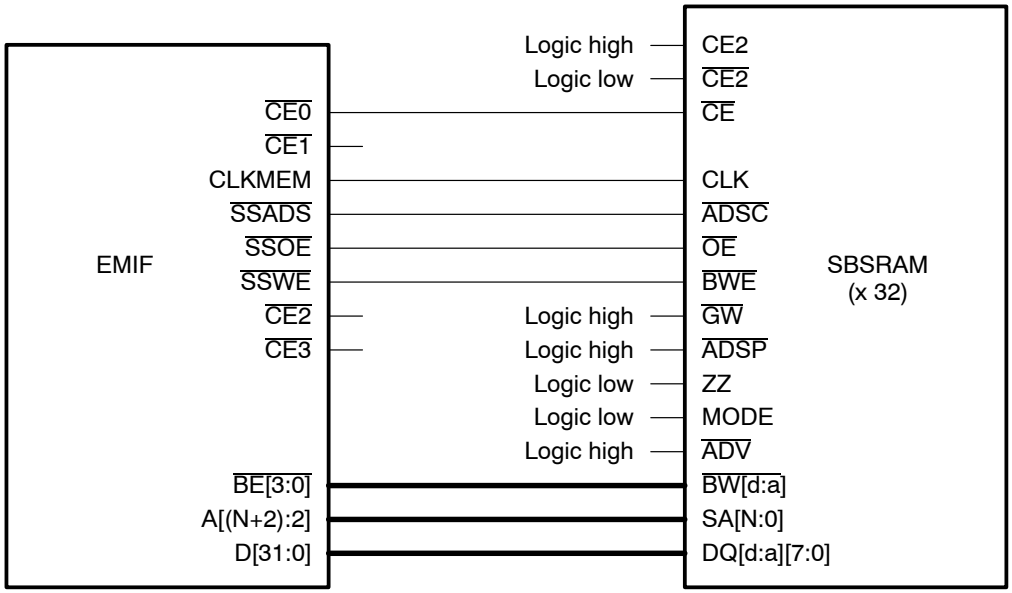
- The EMIF pins $\overline{\text{SSADS}}$, $\overline{\text{SSOE}}$, and $\overline{\text{SSWE}}$ supply the SBSRAM chip with an address strobe signal, an output enable signal, and a write enable signal, respectively. These signals are only valid if the chip select line for the SBSRAM is low.
- The SBSRAM signals $\overline{\text{ADV}}$ and $\overline{\text{ADSP}}$ are not used by the EMIF and thus should be tied high.
- The $\overline{\text{GW}}$ signal provided by the major SBSRAM manufacturers allows all four bytes to be written at once. This feature is not directly supported by EMIF. However, the $\overline{\text{BE}}$[3:0] signals of the EMIF, in conjunction with the $\overline{\text{SSWE}}$ signal of the EMIF, selectively allow one or more bytes to be written at a time to the SBSRAM. For reads, all four $\overline{\text{BE}}$[3:0] signals are driven low.
- The MODE signal of the SBSRAM should be set for Linear Burst.
- For specific information on the SBSRAM signals, refer to the manufacturers' data sheets.

3.2.1 Interfacing to a Non-Parity, Pipeline SBSRAM Chip (32 Bits Wide)

Figure 3–1 shows an example of how to connect the EMIF to a non-parity (x32), pipeline SBSRAM chip. The SBSRAM is enabled in the CE0 space by the EMIF $\overline{\text{CE0}}$ output connected to the SBSRAM $\overline{\text{CE}}$ input. The unused EMIF $\overline{\text{CE}}$ pins ($\overline{\text{CE[3:1]}}$) can be left unconnected, but the unused SBSRAM CE pins and other inputs should be tied off as shown in the figure. SBSRAM signals CE2 and $\overline{\text{CE2}}$ are tied to their active states. The SBSRAM MODE signal is tied low for Linear Burst. All other unused SBSRAM signals (ZZ, $\overline{\text{ADV}}$, $\overline{\text{ADSP}}$, and $\overline{\text{GW}}$) are tied to their inactive states.

The MTYPE field in CE0 space control register1 is set to100b, to indicate SBSRAM.

Figure 3–1. Interface to a Non-Parity, Pipeline SBSRAM Chip (32 Bits Wide)

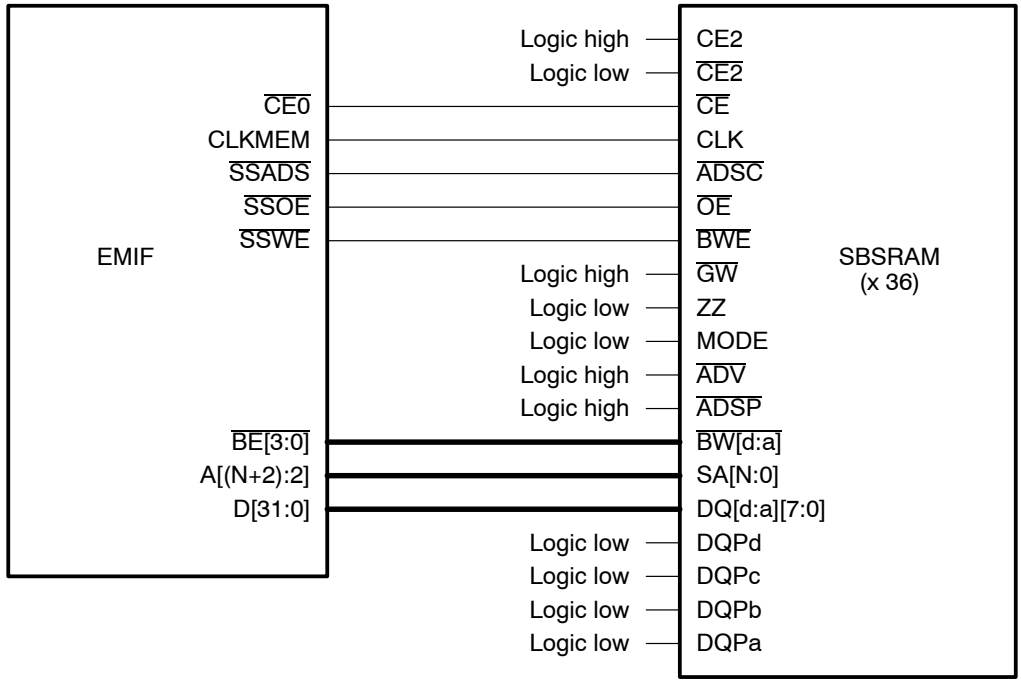


3.2.2 Interfacing to a Parity, Pipeline SBSRAM Chip (36 Bits Wide)

The example in Figure 3–2 shows the EMIF connected to a parity (x36), pipeline SBSRAM chip. The unused parity data bits DQP[d:a] should be tied off as shown to reduce power.

MTYPE = 100b in CE0 space control register1.

Figure 3–2. Interface to a Parity, Pipeline SBSRAM Chip (36 Bits Wide)

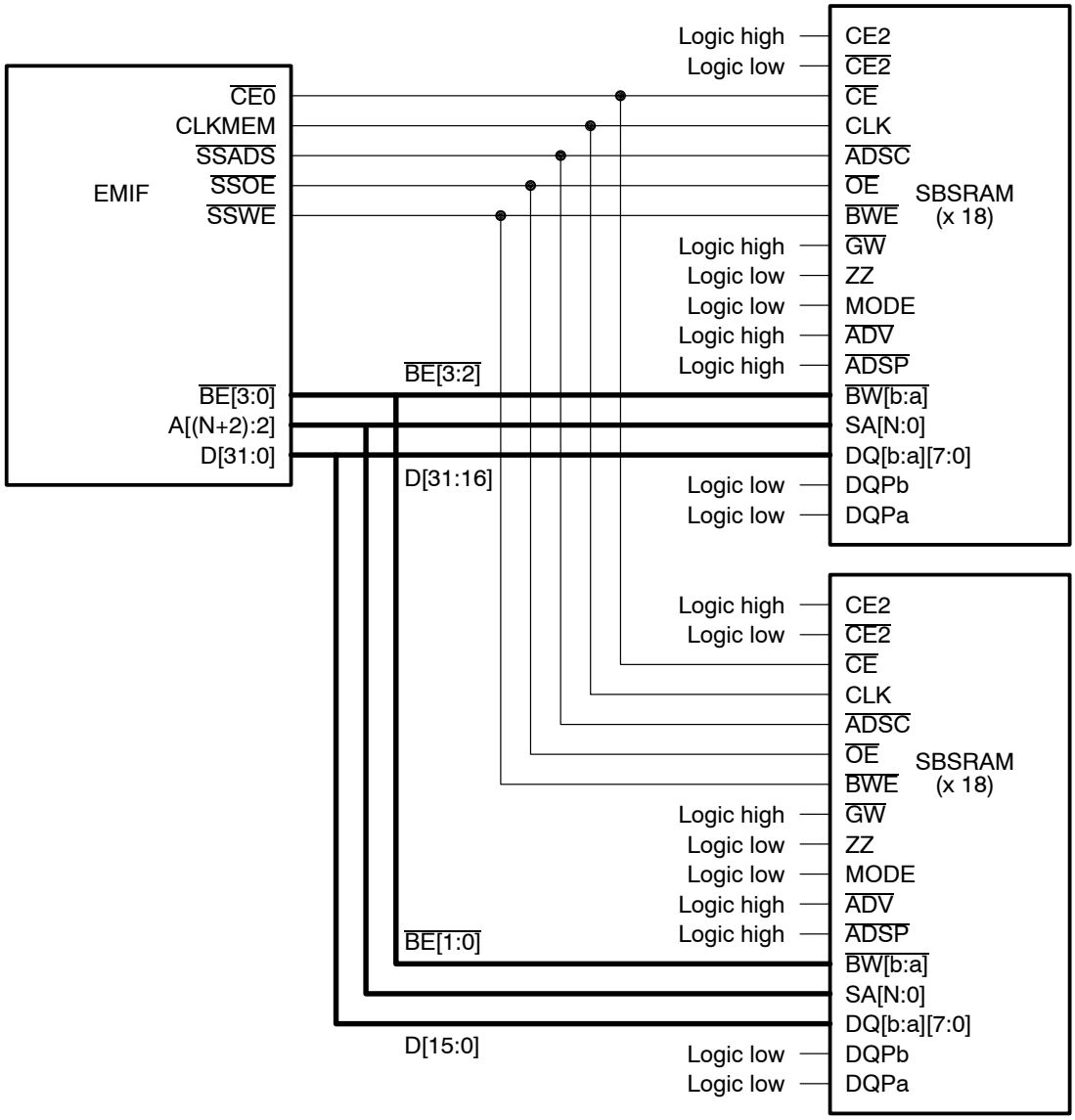


3.2.3 Interfacing to Two Parity, Pipeline SBSRAM Chips (Each 18 Bits Wide)

The example in Figure 3–3 shows the EMIF connected to two parity (x18), pipeline SBSRAM chips. The EMIF data bus and byte enable signals are split between the two memories.

MTYPE = 100b in CE0 space control register1.

Figure 3–3. Interface to Two Parity, Pipeline SBSRAM Chips (Each 18 Bits Wide)



3.3 SBSRAM Power Down Modes

The major SBSRAM manufacturers support a power saving snooze mode. When an asynchronous external pin (ZZ) on the SBSRAM chip is active, the device enters a low-power standby mode. All data in the memory array is maintained, and all other SBSRAM inputs besides ZZ are ignored. For the latency between disabling/enabling of the snooze mode and valid access to the SBSRAM, consult the manufacturer's data sheet. Typically, this latency is up to four clock cycles for both entering and exiting snooze mode. The EMIF does not directly support this capability. However, the enabling/disabling of ZZ can be accomplished with the use of a C55x general purpose I/O pin.

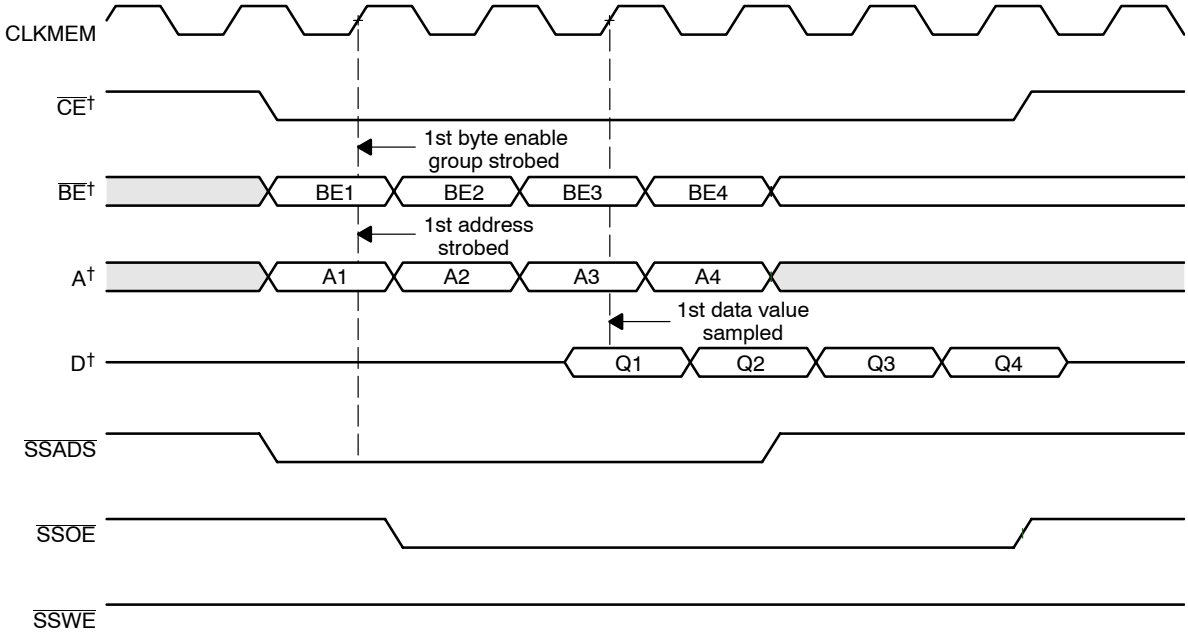
The EMIF does directly support another power down mode of the SBSRAM chips. With SBSRAM chip select lines inactive and $\overline{\text{ADSC}}$ inactive, a reduced power state is achieved. To further reduce power, if CLKMEM is not used for other memories, CLKMEM can be turned off with the MEMCEN bit of the EMIF global control register.

3.4 SBSRAM Read Operations

Figure 3–4 shows signal activity involved when the EMIF performs a DMA burst to read four 32-bit values from an SBSRAM. Every access strobes a new address into the SBSRAM. The period during which the addresses are driven is indicated by the address strobe signal ($\overline{\text{SSADS}}$) being low. Pipeline SBSRAM has a read latency of two cycles. The address is strobed into the SBSRAM on the first rising CLKMEM edge after $\overline{\text{SSADS}}$ goes active, and data is latched by the EMIF 2 cycles later. Although the first access requires a delay of 2 cycles before the data is present on the bus, each subsequent data transfer has single-cycle throughput.

For more detailed timing information, see the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076).

Figure 3–4. SBSRAM Read Operation

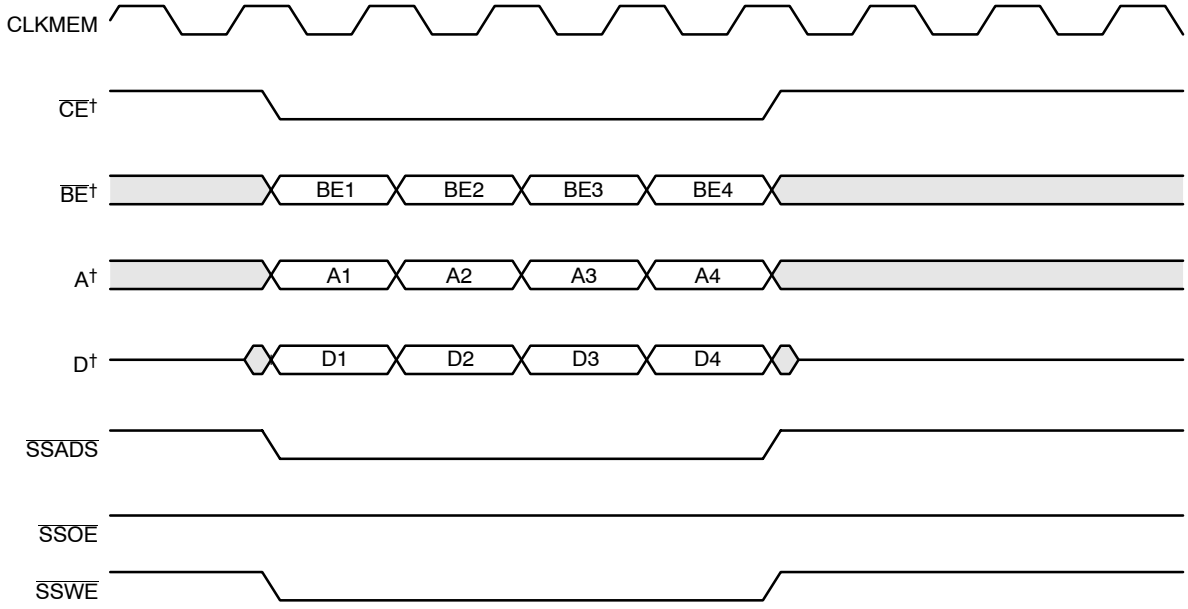


† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.

3.5 SBSRAM Write Operations

Figure 3–5 shows signal activity involved when the EMIF performs a DMA burst to write four 32-bit values to an SBSRAM chip. For more detailed timing information, see the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076).

Figure 3–5. SBSRAM Write Operation



† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.

This page is intentionally left blank.

Chapter 4

Using SDRAM

The EMIF supports a glueless interface to high density and high speed synchronous DRAM (SDRAM). Both 64M-bit and 128M-bit SDRAM chips are supported in 16-bit and also 32-bit data widths. For SDRAM, the EMIF can be configured to operate at 1/2 or 1 times the DSP CPU clock frequency; however, the frequencies you can choose among depend on the timing limitations shown in the electrical specifications section of the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076).

Topic **Page**

4.1 SDRAM Interface Options

Table 4–1 shows the register bit configurations and address pin mapping to row and column addresses for specific organizations and quantities of SDRAM chips. For descriptions of the register configuration bits SDACC, SDSIZE, and SDWID, see the description of the SDRAM control registers in section 5.6 (page 5-16). The pins named in the table are described in section 1.2 (page 1-4).

Table 4–1. EMIF SDRAM Configuration and Address Pin Mapping

SDRAM	Configuration Bits			CE Spaces	Physical Interface	Bank/Row Address	Column Address	See ...
	SDACC	SDSIZE	SDWID					
64M bits: One chip, 4M x 16 bit	0	0	0	CE0–CE1 or CE2–CE3	SDRAM	BA[1:0] and A[11:0]	A[7:0]	Section 4.3.1 on page 4-7
					EMIF	A[14:12], SDA10, and A[10:1]	A[8:1]	
128M bits: Two chips, 4M x 16 bit	1	0	0	All four	SDRAM	B [1:0] and A[11:0]	A[7:0]	Section 4.3.2 on page 4-8
					EMIF	A[15:13], SDA10, and A[11:2]	A[9:2]	
64M bits: One chip, 2M x 32 bit	1	0	1	CE0–CE1 or CE2–CE3	SDRAM	BA[1:0] and A[10:0]	A[7:0]	Section 4.3.3 on page 4-10
					EMIF	A[14:13], SDA10, and A[11:2]	A[9:2]	
64M bits: Two chips, 2M x 32 bit	1	0	1	All four	SDRAM	BA[1:0] and A[10:0]	A[7:0]	Section 4.3.4 on page 4-11
					EMIF	A[14:13], SDA10, and A[11:2]	A[9:2]	
128M bits: One chip, 8M x 16 bit	0	1	0	All four	SDRAM	BA[1:0] and A[11:0]	A[8:0]	Section 4.3.5 on page 4-12
					EMIF	A[14:12], SDA10, and A[10:1]	A[9:1]	
128M bits: One chip, 4M x 32 bit	1	1	1	All four	SDRAM	BA[1:0] and A[11:0]	A[7:0]	Section 4.3.6 on page 4-13
					EMIF	A[15:13], SDA10, and A[11:2]	A[9:2]	

4.2 Configuring the EMIF for SDRAM Accesses

Preparations must be made before the EMIF is used to access external SDRAM. Section 4.2.1 explains which register fields must be programmed. Section 4.2.2 provides a configuration procedure.

4.2.1 Register Fields To Program

To configure the EMIF for interfacing to SDRAM, program the following register fields. The EMIF register descriptions are in Chapter 5.

- Each CE space has a CE space control register 1 that includes a memory type (MTYPE) field. For each CE space that contains SDRAM, define the memory type as SDRAM (MTYPE = 011b).
- The EMIF contains one pair of SDRAM control registers for configuring parameters that apply to *all CE spaces containing SDRAM*. Three fields of these registers define characteristics the EMIF must know to interface properly with the SDRAM chip or chips:
  - SDACC of SDRAM control register 2. This bit defines the EMIF data bus width (16 bits or 32 bits) that will be used to perform the memory access.
  - SDSIZE of SDRAM control register 1. This bit indicates the size of each SDRAM chip, either 64M bits or 128M bits.
  - SDWID of SDRAM control register 1. This bit indicates the width of the data bus on each SDRAM chip, either 16 bits wide or 32 bits wide.
- The SDRAM control registers also contain fields for setting standard timing parameters. For the settings required for a particular SDRAM chip, consult the SDRAM manufacturer's documentation.
- SDRAM control register 1 includes the refresh enable (RFEN) bit. Set this bit if the EMIF will be responsible for SDRAM refreshes.
- The SDRAM period register defines the trickle refresh period. This period is the time between periodic SDRAM refreshes initiated by the EMIF when RFEN = 1.
- The EMIF global control register has two fields for controlling the memory clock signal, which is used for all SDRAM and all SBSRAM:
  - MEMCEN. This bit enables or disables the memory clock on the CLKMEM pin.
  - MEMFREQ. When the memory clock is disabled (MEMCEN = 0), write to MEMFREQ to select the relative frequency of the memory clock signal. After programming MEMFREQ and the DSP clock generator, enable the clock (MEMCEN = 1).

- The EMIF global control register also has two fields that affect all CE spaces, regardless of memory type:
  - WPE. This bit enables or disables write posting (see section 1.7 on page 1-14) for all CE spaces.
  - NOHOLD. This bit enables or disables HOLD requests (see section 1.6 on page 1-13) for all CE spaces.
- The SDRAM initialization register is used to initialize all SDRAM. Any write to this register causes an SDRAM initialization sequence (see section 4.7 on page 4-17). After a hardware reset or after powering up the C55x device, a write to this register should be performed following the configuration of all CE spaces and prior to the accessing of SDRAM.

4.2.2 Configuration Procedure

Before configuring the EMIF, make sure the DSP clock generator is properly programmed. Its output is divided according to the MEMFREQ field in EGCR to produce the memory clock. The frequency of the memory clock must meet (1) the timing requirements in the SDRAM manufacturer's documentation and (2) the timing limitations shown in the electrical specifications section of the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076).

Use the following procedure to prepare the EMIF for accessing external SDRAM:

1) Clear the MEMCEN bit in the EMIF global control register (EGCR). This prevents the memory clock from being driven on the CLKMEM pin.
2) Keeping MEMCEN = 0, program the other programmable fields of EGCR: MEMFREQ, WPE, NOHOLD, and (if asynchronous memory is used in any of the CE spaces) ARDYOFF.
3) For each CE space that contains SDRAM, set MTYPE = 011b in CE space control register 1.
4) Program SDRAM control register 1.
5) Program SDRAM control register 2.
6) If the EMIF is responsible for refreshing the SDRAM, write the desired refresh period to the SDRAM period register.

7) Set the MEMCEN bit in EGCR, so that the memory clock is driven on the CLKMEM pin.

8) Write to the SDRAM initialization register. After new values are written to EMIF configuration registers, 6 CPU clock cycles are required for the new configuration to propagate through the EMIF logic. After this delay, the EMIF begins the SDRAM initialization sequence.

4.3 Examples of Interfacing to SDRAM Chips

The following examples are provided to help you to connect pins and to configure the EMIF for various arrangements of SDRAM chips. For details about the EMIF pins shown in the examples, see section 1.2 on page 1-4.

4.3.1 Interfacing to a 4M x 16 SDRAM Chip

Figure 4–1 and Table 4–2 provide an example of interfacing to 64M-bit SDRAM chip with a 4M x 16 configuration. Since the limit for a single CE space is 32M bits, two CE spaces are used. However, only a single $\overline{CE}$ pin corresponding to the beginning of the mapped range is required for use as a chip select for the SDRAM.

The CE spaces used must be the first two or last two CE spaces. Specifically, you can use the CE0 and CE1 spaces or the CE2 and CE3 spaces, but you cannot use the CE1 and CE2 spaces.

The example in this section uses the CE0 and CE1 spaces. The only chip enable pin that is used is $\overline{CE0}$. $\overline{CE1}$ is left unconnected. The other $\overline{CE}$ pins ($\overline{CE2}$ and $\overline{CE3}$) can be left unconnected or used for other memory chips (asynchronous or synchronous).

If you were to use the CE2 and CE3 spaces, $\overline{CE2}$ would be used for chip selection. $\overline{CE3}$ would be left unconnected, and $\overline{CE0}$ and $\overline{CE1}$ could be left unconnected or used for other memory chips (asynchronous or synchronous).

For both of the two CE spaces used for SDRAM, the MTYPE bits must be set to 011b.

Figure 4−1. Interfacing to a 64M-bit SDRAM Chip With a 4M x 16 Array Organization

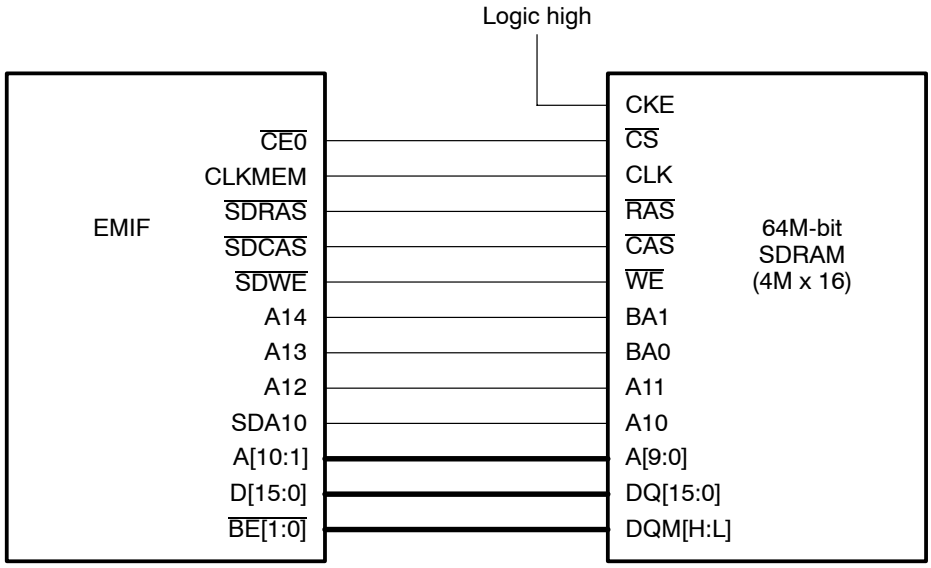


Table 4−2. EMIF Configuration for a 64M-bit SDRAM Chip With a 4M x 16 Array Organization

	Configuration Bits				
SDRAM	**SDACC**	**SDSIZE**	**SDWID**	**MTYPE†**	**CE Spaces Used**
64M bits: One chip, 4M x 16 bit	0	0	0	011b	CE0 and CE1 or CE2 and CE3

† Each CE space has its own MTYPE bits. MTYPE must be 011b for each CE space that is used for SDRAM.

4.3.2 Interfacing to Two 4M x 16 SDRAM Chips

An example of interfacing to two 64M-bit (4M x 16) SDRAM chips is given in Figure 4−2 and Table 4−3. Because the total memory is 128M bits, this memory configuration will occupy all four 32M-bit CE spaces with some possible limitations at the bottom and top of the memory map. See the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076) for device-specific details on these limitations. Only a single $\overline{CE}$ pin ($\overline{CE0}$) is required for use as a chip select for both SDRAM chips. The other $\overline{CE}$ pins ($\overline{CE}[3:1]$) must be left unconnected.

For this SDRAM arrangement, the MTYPE bits for all CE spaces must be set to 011b.

Figure 4–2. Interfacing to Two 64M-bit SDRAM Chips, Each With a 4M x 16 Array Organization

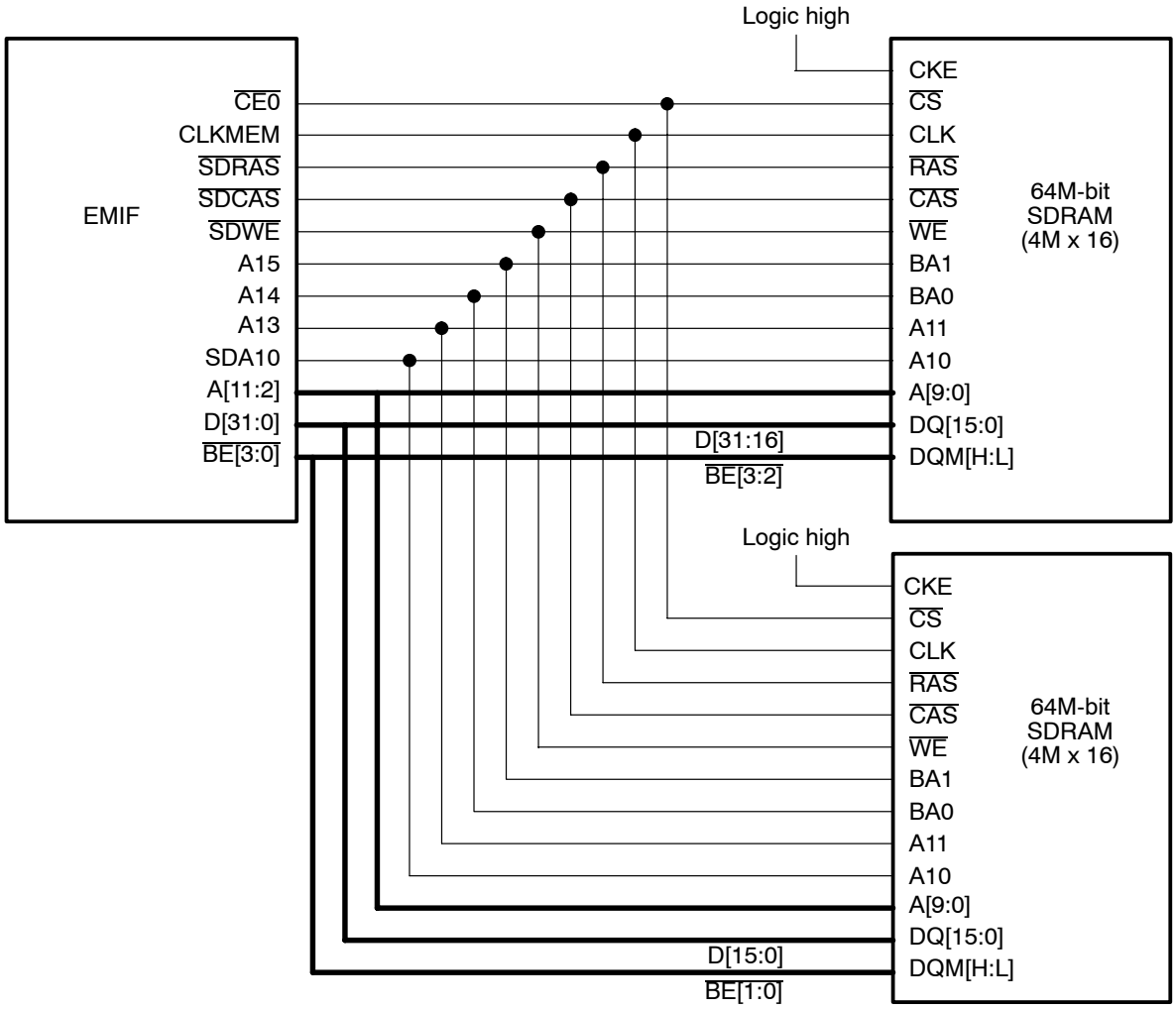


Table 4–3. EMIF Configuration for Two 64M-bit SDRAM Chips, Each With a 4M x 16 Array Organization

	Configuration Bits				
SDRAM	**SDACC**	**SDSIZE**	**SDWID**	**MTYPE†**	**CE Spaces Used**
128M bits: Two chips, each 4M x 16 bit	1	0	0	011b	All four

† Each CE space has its own MTYPE bits. MTYPE must be 011b for each CE space that is used for SDRAM.

4.3.3 Interfacing to a 2M x 32 SDRAM Chip

The example shown in Figure 4–3 and Table 4–4 uses a single 64M-bit (2M x 32) SDRAM. Two contiguous CE spaces are required. This example uses the CE0 and CE1 spaces, with the $\overline{\text{CE0}}$ pin connected to the SDRAM and the $\overline{\text{CE1}}$ pin left unconnected. The pins $\overline{\text{CE2}}$ and $\overline{\text{CE3}}$ can be used for other memory chips.

It is possible to use the CE2 and CE3 spaces instead. The $\overline{\text{CE2}}$ pin would be used, and the $\overline{\text{CE3}}$ pin would be left unconnected. The pins $\overline{\text{CE0}}$ and $\overline{\text{CE1}}$ could be used for other memory chips.

For both of the CE spaces used for SDRAM, the MTYPE bits must be set to 011b.

Figure 4–3. Interfacing to a 64M-bit SDRAM Chip With a 2M x 32 Array Organization

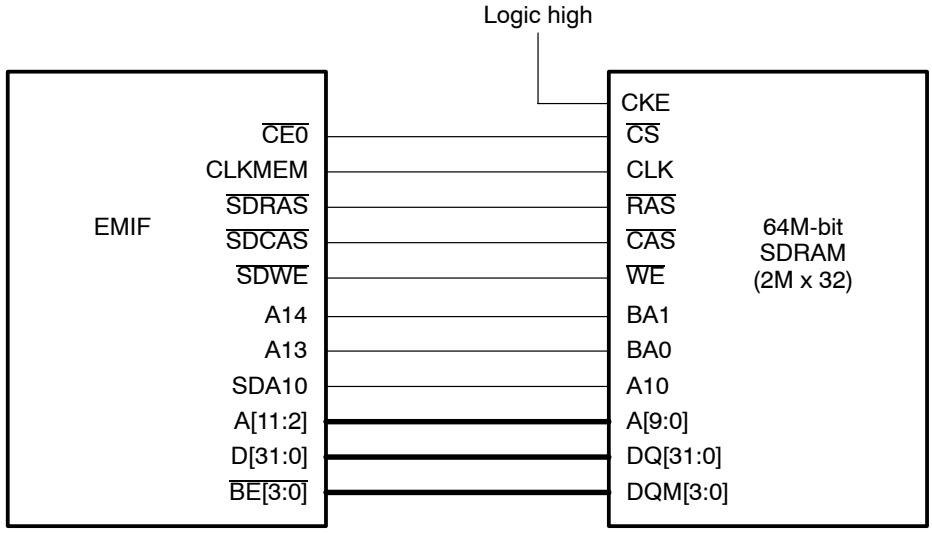


Table 4–4. EMIF Configuration for a 64M-bit SDRAM Chip With a 2M x 32 Array Organization

	Configuration Bits				
SDRAM	**SDACC**	**SDSIZE**	**SDWID**	**MTYPE†**	**CE Spaces Used**
64M bits: One chip, 2M x 32 bit	1	0	1	011b	CE0 and CE1 or CE2 and CE3

† Each CE space has its own MTYPE bits. MTYPE must be 011b for each CE space that is used for SDRAM.

4.3.4 Interfacing to Two 2M x 32 SDRAM Chips

An example of two 64M-bit (2M x 32) SDRAM chips connected to the EMIF is given in Figure 4−4 and Table 4−5. Each of the two SDRAM chips requires the use of two CE spaces. Only one $\overline{CE}$ pin is needed per chip ($\overline{CE0}$ and $\overline{CE2}$). The unneeded $\overline{CE}$ pins ($\overline{CE1}$ and $\overline{CE3}$) must be left unconnected.

For this SDRAM arrangement the MTYPE bits for all CE spaces must be set to 011b.

Figure 4−4. Interfacing to Two 64M-bit SDRAM Chips, Each With a 2M x 32 Array Organization

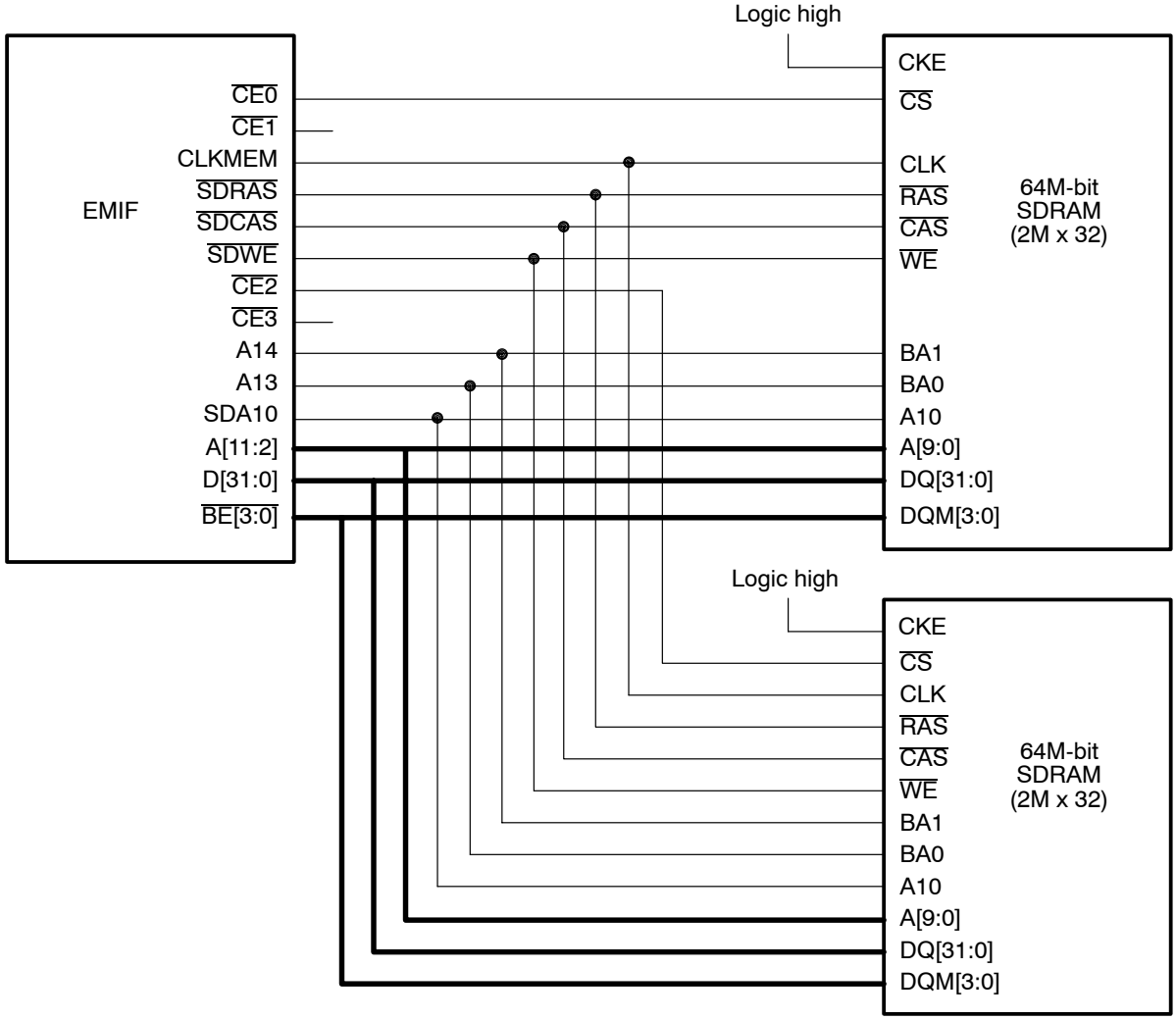


Table 4–5. EMIF Configuration for Two 64M-bit SDRAM Chips, Each With a 2M x 32 Array Organization

SDRAM	Configuration Bits				CE Spaces Used
	SDACC	SDSIZE	SDWID	MTYPE†	
128M bits: Two chips, each 2M x 32 bit	1	0	1	011b	All four

† Each CE space has its own MTYPE bits. MTYPE must be 011b for each CE space that is used for SDRAM.

4.3.5 Interfacing to an 8M x 16 SDRAM Chip

Figure 4–5 shows a single 128M-bit (8M x 16) SDRAM chip interfaced to the EMIF. Table 4–6 lists the required configuration bit values for this setup. All CE spaces are occupied. $\overline{CE0}$ used as the SDRAM chip select pin, and $\overline{CE[3:1]}$ are left unconnected.

For this SDRAM arrangement, the MTYPE bits for all CE spaces must be set to 011b.

Figure 4–5. Interfacing to a 128M-bit SDRAM Chip With an 8M x 16 Array Organization

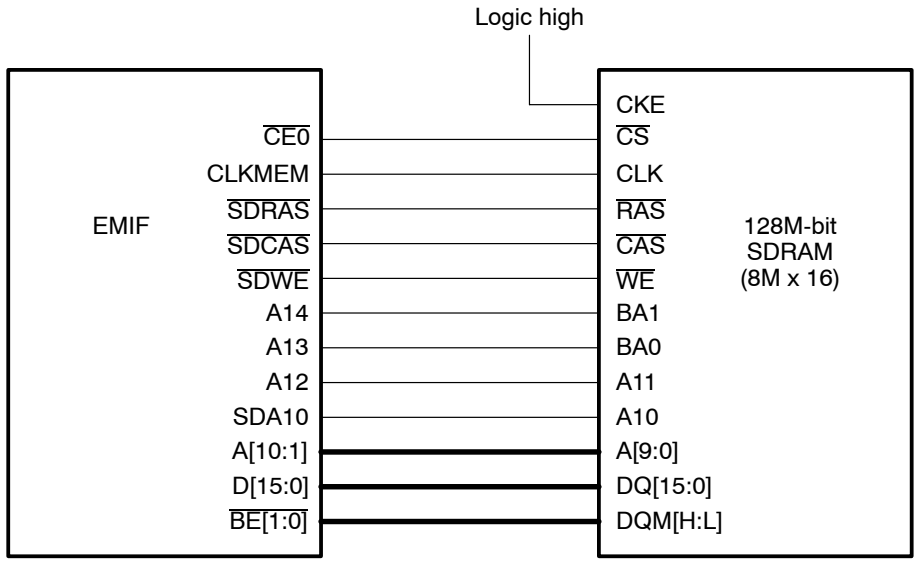


Table 4–6. EMIF Configuration for a 128M-bit SDRAM Chip With an 8M x 16 Array Organization

	Configuration Bits				
SDRAM	**SDACC**	**SDSIZE**	**SDWID**	**MTYPE†**	**CE Spaces Used**
128M bits: One chip, 8M x 16 bit	0	1	0	011b	All four

† Each CE space has its own MTYPE bits. MTYPE must be 011b for each CE space that is used for SDRAM.

4.3.6 Interfacing to a 4M x 32 SDRAM Chip

A single 128M-bit (4M x 32) SDRAM chip interfaced to the EMIF is shown in Figure 4–6. The required configuration bit values are shown in Table 4–7. All CE spaces are occupied. $\overline{CE0}$ is used as the SDRAM chip select pin, and $\overline{CE}[3:1]$ are left unconnected.

For this SDRAM arrangement, the MTYPE bits for all CE spaces must be set to 011b.

Figure 4–6. Interfacing to a 128M-bit SDRAM Chip With a 4M x 32 Array Organization

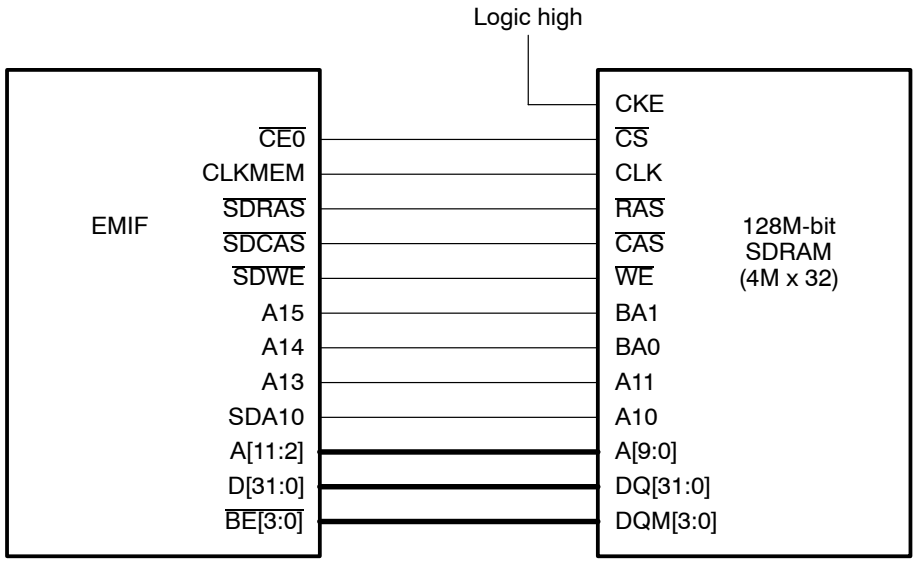


Table 4–7. EMIF Configuration for a 128M-bit SDRAM Chip With a 4M x 32 Array Organization

	Configuration Bits				
SDRAM	**SDACC**	**SDSIZE**	**SDWID**	**MTYPE†**	**CE Spaces Used**
128M bits: One chip, 4M x 32 bit	1	1	1	011b	All four

† Each CE space has its own MTYPE bits. MTYPE must be 011b for each CE space that is used for SDRAM.

4.4 SDRAM Commands

The EMIF provides support for industry standard SDRAM commands. These commands and their corresponding pin states are given in Table 4–8 and Table 4–9, respectively.

Table 4–8. SDRAM Commands

Command	Description
DCAB	Deactivate (pre-charge) all banks
ACTV	Activate the selected bank and select the row
READ	Input the starting column address and begin the read operation
WRT	Input the starting column address and begin the write operation
MRS	Set the mode register of the SDRAM chip(s)
REFR	Auto refresh cycle with internal address
NOP	No SDRAM operation

Table 4–9. EMIF Pin States for the SDRAM Commands

Command	$\overline{\text{CEn}}$	$\overline{\text{SDRAS}}$	$\overline{\text{SDCAS}}$	$\overline{\text{SDWE}}$	Higher Address Lines†	SDA10	Lower Address Lines†
DCAB	0	0	1	0	X	1	X
ACTV	0	0	1	1	ROW	ROW	ROW
READ	0	1	0	1	ROW	0	COL
WRT	0	1	0	0	ROW	0	COL
MRS	0	0	0	0	X	X	MRS
REFR	0	0	0	1	X	X	X
NOP	0	1	1	1	X	X	X
	1	X	X	X	X	X	X

† As shown in Table 4–1 on page 4-3, the number and type of SDRAM chips determine which address lines are used.

4.5 Address Shift Performed by the EMIF

Because the same EMIF pins address both the row and column addresses, the EMIF appropriately shifts the address for proper row and column address selection. SDRAM chips use the A10 inputs for control as well as address. The EMIF forces the precharge disable bit (SDA10) to be low during READ and WRT commands, high during DCAB commands, and as bit 10 of the row address during ACTV commands. This prevents the auto-precharge from occurring following a READ or WRT command.

4.6 Monitoring SDRAM Page Boundaries

Because SDRAM is a paged memory type, the EMIF SDRAM controller monitors the active row of SDRAM so that row boundaries are not crossed during accesses. The EMIF stores the address of the open page and performs a comparison against that address for subsequent accesses to the SDRAM bank. This storage and comparison is performed separately for each CE space. The number of address bits and page boundaries compared are a function of the particular SDRAM interface option selected. For the options, see section 4.1 on page 4-2.

During the course of an access, if a page boundary is crossed, the EMIF performs a DCAB command and starts a new row access (ACTV command). Simply ending the current access is not a condition which forces the active SDRAM row to be closed. The EMIF speculatively leaves the active row open until it becomes necessary to close it. This feature decreases the DCAB ACTV overhead and improves the performance of the interface. However, the EMIF only allows one row to be open at a time regardless of the number of banks that the SDRAM contains.

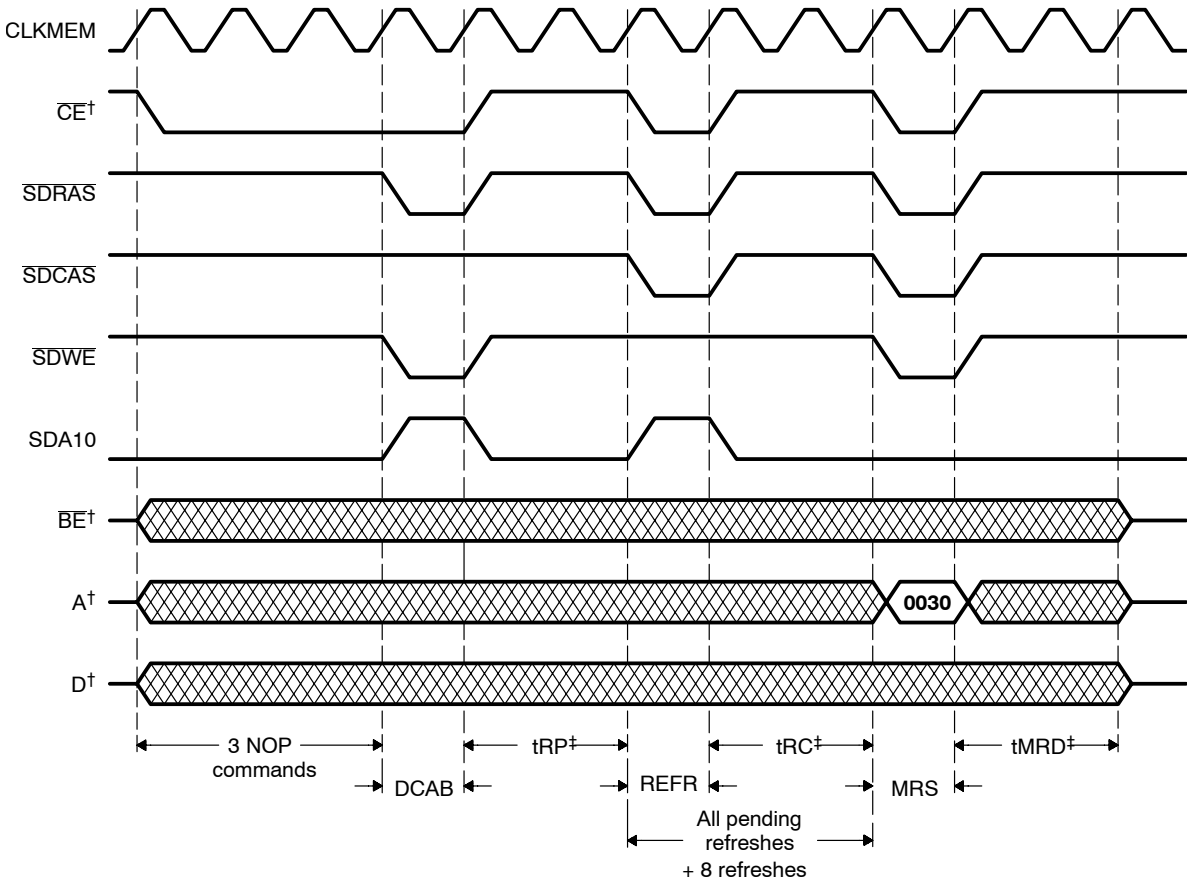
4.7 SDRAM Initialization

For all CE spaces configured as SDRAM, the EMIF performs the necessary functions to initialize SDRAM. An SDRAM initialization is requested by any write to the EMIF SDRAM initialization register. The MTYPE fields of the configuration registers should be properly set before performing the initialization. Figure 4–7 shows the timing during SDRAM initialization and execution of the MRS command.

The SDRAM initialization sequence is as follows:

1) Three NOP commands are sent to all CE spaces configured as SDRAM.
2) A DCAB command is sent to all CE spaces configured as SDRAM.
3) Eight REFR commands are sent to all CE spaces configured as SDRAM.
4) An MRS command is sent to all CE spaces configured as SDRAM.
5) The SDRAM initialization register is cleared to prevent multiple MRS cycles.

Figure 4–7. SDRAM Initialization, Including a Mode Register Set (MRS) Command



† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.
‡ The timing parameters tRP, tRC, and tMRD must be programmed via the SDRAM control registers (see section 5.6 on page 5-16).

4.8 SDRAM Mode Register Set

Each SDRAM chip contains a mode register that dictates the operating characteristics of the SDRAM chip. The EMIF automatically performs a DCAB command, followed by a MRS command during SDRAM initialization. Like other SDRAM commands generated by the EMIF, MRS commands are sent to all CE spaces configured as SDRAM. The EMIF always uses a mode register value of 0x0030 during a MRS command. Figure 4–8 shows the mapping between mode register bits, EMIF pins, and the mode register value. Table 4–10 shows the standard SDRAM configuration values selected by the mode register value that is sent by the EMIF.

Figure 4–8. SDRAM Mode Register Value Sent by the EMIF to SDRAM

Mode Register Bits:	11	10	9	8	7	6	5	4	3	2	1	0
EMIF Pins:	A13	SDA10	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
Field Description:	Reserved		Write burst length	Reserved		Read Latency			Serial/ interleave burst type	Burst length		
Value:	0	0	0	0	0	0	1	1	0	0	0	0

Table 4–10. SDRAM Configuration Sent With the MRS Command

Field	Selection
Write burst length	1
Read latency	3
Serial/interleave burst type	0 (serial)
Burst length	1

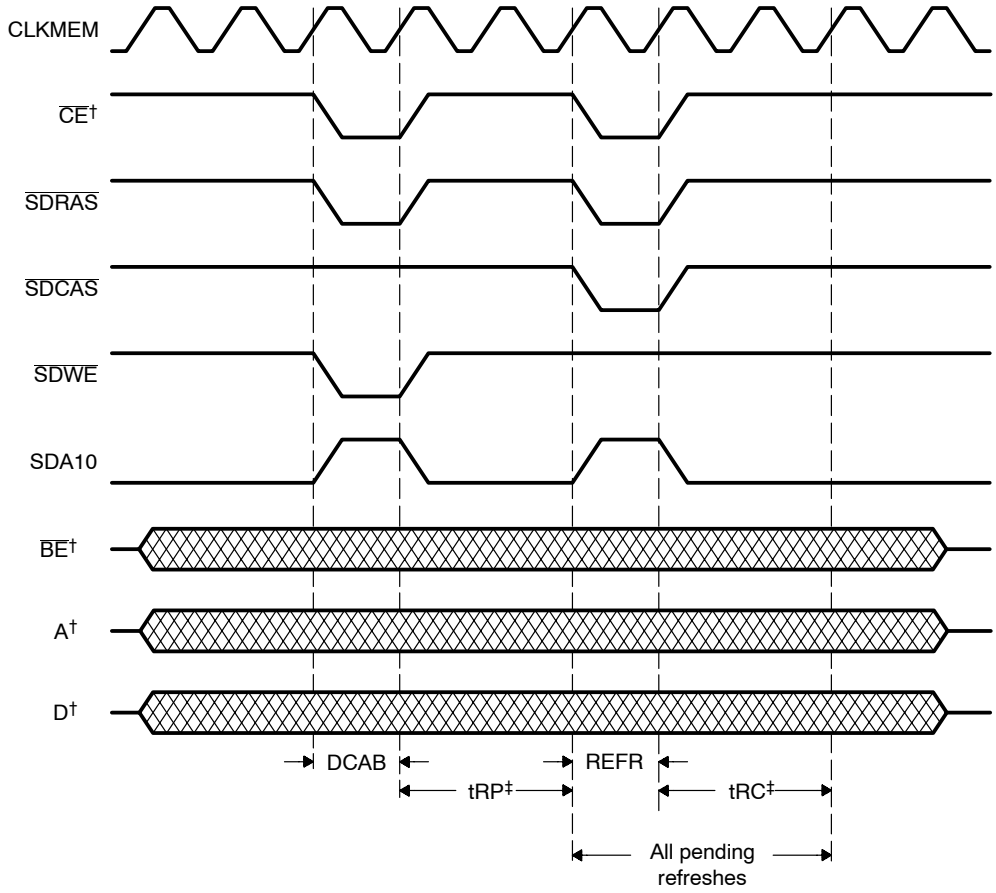
4.9 SDRAM Refresh

The EMIF can send refresh (REFR) commands to SDRAM. The RFEN bit in SDRAM control register 1 (SDC1) selects or deselects the SDRAM refresh mode of the EMIF. A value of 0 in RFEN disables all EMIF-initiated refreshes, and you must ensure that refreshes are implemented in an external device. A value of 1 in RFEN enables the EMIF to send REFR commands. The timing diagram for an SDRAM refresh is shown in Figure 4–9.

Section 4.9.1 explains when the EMIF sends refresh commands. When the command is sent, it is sent to all CE spaces that are configured for SDRAM with the corresponding MTYPE bits. REFR is automatically preceded by a DCAB command. This ensures that all CE spaces configured for SDRAM are deactivated. Page information is always invalid before and after a REFR command. Therefore, a refresh cycle always forces a page miss on the next access.

Section 4.9.2 explains why the EMIF cannot send refresh commands with some idle modes of the DSP.

Figure 4–9. DCAB-REFR Command Sequence Sent by EMIF to SDRAM



† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.
‡ The timing parameters tRP and tRC must be programmed via the SDRAM control registers (see section 5.6 on page 5-16).

4.9.1 Refresh Requests Generated Within the EMIF

Refresh requests are generated by two counters inside the EMIF. The EMIF monitors the number of refresh requests generated and performs the refreshes when it is not servicing higher priority requests. For a list of requesters and their relative priorities, see section 1.3 on page 1-9.

One of the counters generates periodic *trickle refresh requests*. You define the period between these requests by writing the desired count to the 16-bit SDRAM period register (SDPER). The counter is decremented every CLKMEM cycle. When counter reaches 0, it is reloaded from SDPER and then continues to be decremented. You can read the SDRAM counter register (SDCNT) to get the current value in the refresh counter.

The other counter inside the EMIF generates *urgent refresh requests*. This is a 2-bit counter that records the backlog of trickle refresh requests. It is incremented by 1 for each trickle refresh request generated and decremented by 1 for each refresh performed. The counter saturates at the values of 11b and 00b. If the counter is 11b, an urgent refresh request is generated.

In response to an urgent refresh request, the EMIF closes the current SDRAM page. Following a DCAB command, the EMIF SDRAM controller performs three REFR commands (thereby decrementing the counter to 00b) before proceeding with the remainder of the current access. The DCAB-REFR command sequence occurs in all CE spaces that contain SDRAM. At reset, the 2-bit counter is automatically set to 11b to ensure that several refreshes occur before accesses begin.

During times of inactivity on the SDRAM pins, if no SDRAM access request is pending, the EMIF sends REFR commands as long as the 2-bit counter value is nonzero. This feature reduces the likelihood of having to perform urgent refreshes during actual SDRAM accesses. This refresh occurs only if all SDRAM chips are inactive with invalid page information.

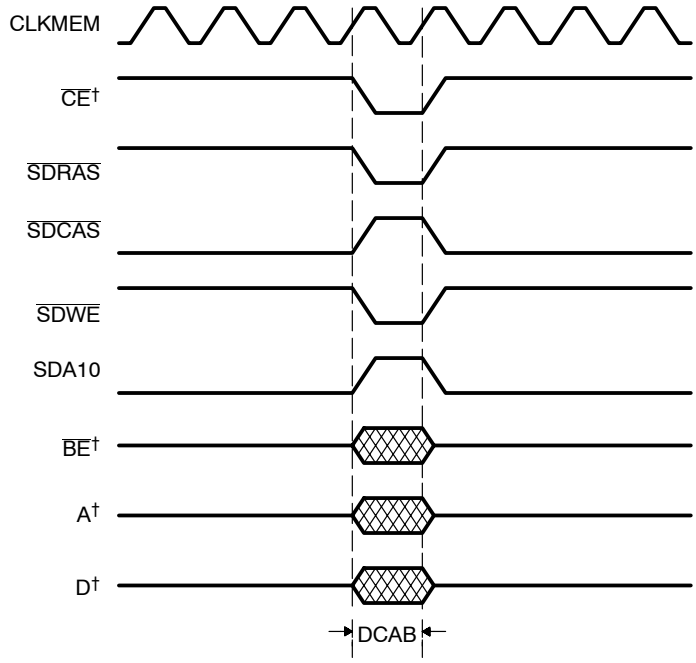
4.9.2 EMIF Cannot Perform Refreshes With Some Idle Modes

If the EMIF or the clock generator is placed into its idle mode by software, the SDRAM clock is disabled to save additional power. The EMIF does not issue SDRAM refresh commands, and the data stored in SDRAM is lost. If an SDRAM refresh were required during this time, external refresh logic would be necessary.

4.10 SDRAM Deactivation

The SDRAM deactivation (DCAB) or precharge command is generated by the EMIF to close the active page of memory. DCAB is performed after hardware reset or after a write to the SDRAM initialization register (INIT). Also, a DCAB is generated prior to REFR, MRS, and when page boundaries are crossed. During the DCAB command operation, SDA10 is driven high to ensure that all SDRAM banks are deactivated. Figure 4−10 shows the timing diagram for SDRAM deactivation.

Figure 4−10. SDRAM Deactivation



† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.

4.11 SDRAM Read Operations

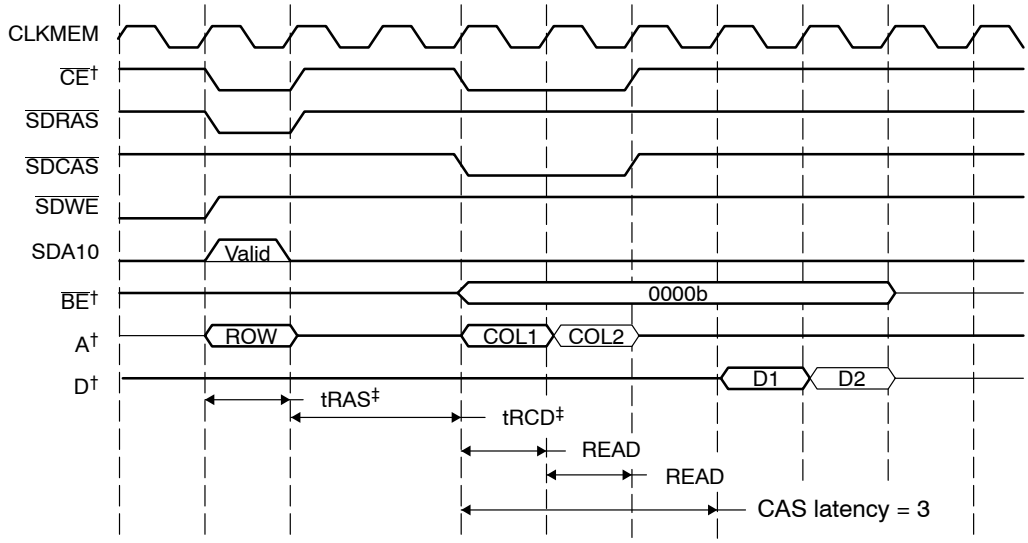
During an SDRAM read, the selected bank is activated with the row address during the ACTV command. The EMIF uses a CAS latency of 3 and a burst length of 1. The 3-cycle latency causes read data to appear on the data bus 3 cycles after the corresponding column address.

If a refresh cycle or an access to a different page of memory is required, a DCAB cycle is performed to deactivate the bank after the last column access. Delay cycles are inserted between the final read command and the DCAB command to meet SDRAM timing requirements. Note that due to the data latency, the transfer of data actually completes after the DCAB command. If no new access is pending, the DCAB command is not performed until such time that the page information becomes invalid.

An example of two, back-to-back SDRAM reads is shown in Figure 4–11.

If there is a CE space change or a direction change (read to write, or write to read) between two consecutive requests, the EMIF inserts one dead cycle. During this dead cycle all chip enable signals are driven high (inactive).

Figure 4–11. SDRAM Read Operation



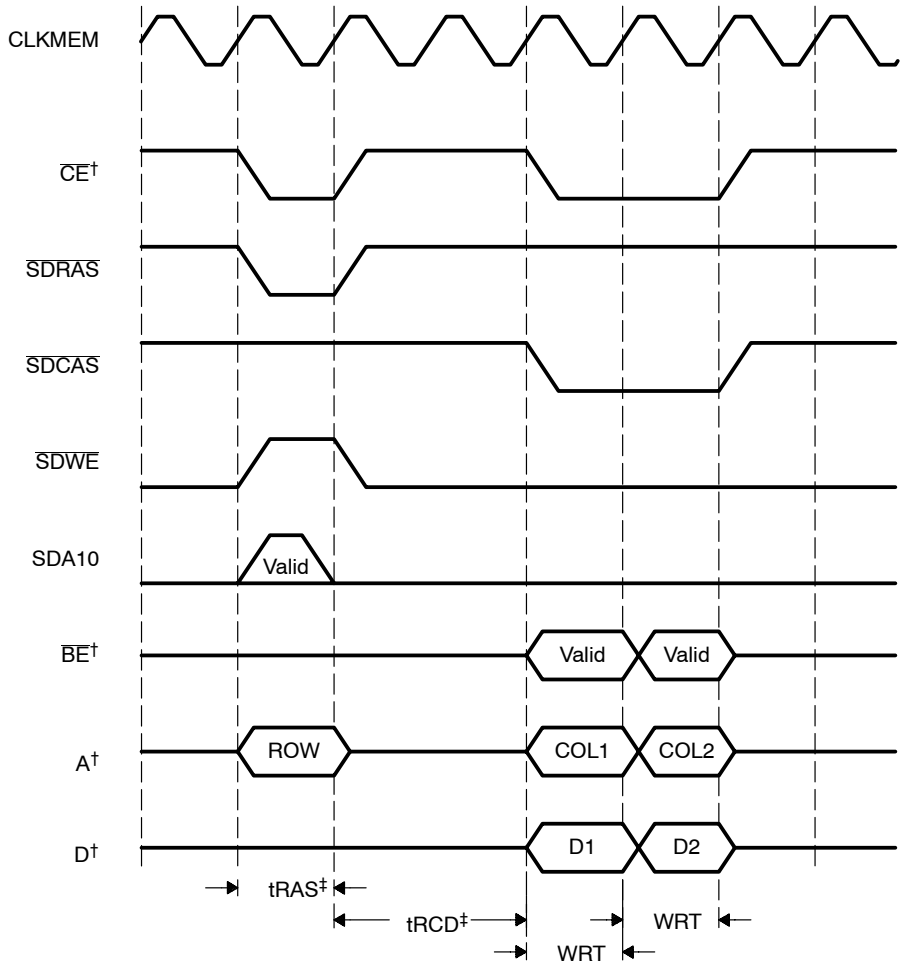
† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.
‡ The timing parameters tRAS and tRCD must be programmed via the SDRAM control registers (see section 5.6 on page 5-16).

4.12 SDRAM Write Operations

All SDRAM writes have a burst length of 1. The bank is activated with the row address during the ACTV command. Unlike SDRAM reads, there is no CAS latency on writes. Therefore, data is output on the same cycle as the column address. Byte and word writes are enabled via the appropriate $\overline{BE}[3:0]$ pins of the EMIF; connect these pins to the DQM inputs of the SDRAM. Following the final write command, delay cycles are inserted to meet SDRAM timing requirements. If required, the bank is then deactivated with a DCAB command and the memory interface can begin a new page access. If no new access is pending or if an access is pending to the same page, the DCAB command is not performed until such time that the page information becomes invalid. Examples of SDRAM writes are shown in Figure 4–12.

If there is a CE space change or a direction change (read to write, or write to read) between two consecutive requests, the EMIF inserts one dead cycle. During this dead cycle all chip enable signals are driven high (inactive).

Figure 4−12. SDRAM Write Operation



† The chip enable, byte enable, address, and data pins used depend on the specific access and EMIF configuration.

‡ The timing parameters tRAS and tRCD must be programmed via the SDRAM control registers (see section 5.6 on page 5-16).

Chapter 5

EMIF Registers

This chapter describes the contents and function of each of the EMIF registers.

Notes:

1) Do not write to configuration registers when external memory accesses are in progress or pending in the EMIF.

2) After new values are written to configuration registers, allow at least 6 CPU clock cycles for the new configuration to propagate through the EMIF logic.

Topic **Page**

5.1 Summary of the EMIF Registers

Table 5–1 lists the addresses and names of the EMIF registers.

Table 5–1. Registers of the EMIF

I/O Address	Register(s)	Description	See ...
0800h	EGCR	EMIF global control register	Page 5-3
0801h	EMIRST	EMIF global reset register	Page 5-6
0802h	EMIBE	EMIF bus error status register	Page 5-7
0803h	CE01	CE0 space control register 1	Page 5-11
0804h	CE02	CE0 space control register 2	Page 5-11
0805h	CE03	CE0 space control register 3	Page 5-11
0806h	CE11	CE1 space control register 1	Page 5-11
0807h	CE12	CE1 space control register 2	Page 5-11
0808h	CE13	CE1 space control register 3	Page 5-11
0809h	CE21	CE2 space control register 1	Page 5-11
080Ah	CE22	CE2 space control register 2	Page 5-11
080Bh	CE23	CE2 space control register 3	Page 5-11
080Ch	CE31	CE3 space control register 1	Page 5-11
080Dh	CE32	CE3 space control register 2	Page 5-11
080Eh	CE33	CE3 space control register 3	Page 5-11
080Fh	SDC1	SDRAM control register 1	Page 5-16
0810h	SDPER	SDRAM period register	Page 5-19
0811h	SDCNT	SDRAM counter register	Page 5-19
0812h	INIT	SDRAM initialization register	Page 5-21
0813h	SDC2	SDRAM control register 2	Page 5-16

5.2 EMIF Global Control Register (EGCR)

The global control register (see Figure 5–1) is a 16-bit I/O-mapped register used to configure and monitor global conditions in the EMIF. Use this register to set up the clock for synchronous memory chips (MEMFREQ and MEMCEN), to enable or disable write posting (WPE), to monitor certain EMIF pins (ARDY, HOLD, and HOLDA), to disable the ARDY signal (ARDYOFF), and to allow or disallow HOLD requests (NOHOLD). Table 5–2 describes the bit fields of EGCR.

Figure 5–1. EMIF Global Control Register (EGCR)

15	14	13	12	11	10	9	8
Reserved				Reserved†	MEMFREQ		Reserved†
R–0				R/W–0	R/W–0		R/W–0
7	6	5	4	3	2	1	0
WPE	Reserved†	MEMCEN	ARDYOFF	ARDY	HOLD	HOLDA	NOHOLD
R/W–0	R/W–0	R/W–1	R/W–0	R–pin	R–pin	R–0	R/W–0

Legend: R = Read; W = Write; -*n* = Value after reset; –pin = Value after reset depends on the associated pin
† Always write 0 to these reserved bits.

Table 5–2. EMIF Global Control Register (EGCR) Bits

Bit	Field	Value	Description
15–12	Reserved		These read-only reserved bits return 0s when read.
11	Reserved	0	Always write 0 to this reserved bit.
10–9	MEMFREQ		Memory clock frequency bits. The CLKMEM pin of the EMIF provides the clock signal for synchronous memory (SBSRAM or SDRAM) chips. MEMFREQ determines the relationship between the frequency of the CPU clock signal and the frequency of the signal on the CLKMEM pin. **Note 1:** The frequency you can use depends on the timing limitations shown in the electrical specifications section of the *TMS320VC5510 Fixed-Point Digital Signal Processor Data Manual* (SPRS076). **Note 2:** Change MEMFREQ only when CLKMEM is disabled (when the MEMCEN bit is 0). To enable CLKMEM, set the MEMCEN bit.
		00b	The CLKMEM frequency is equal to the CPU clock frequency.
		01b	The CLKMEM frequency is 1/2 the CPU clock frequency.
		other	Reserved (do not use)
8	Reserved		This read-only reserved bit returns 0 when read.

Table 5–2. EMIF Global Control Register (EGCR) Bits (Continued)

Bit	Field	Value	Description
7	WPE		Write posting enable bit. Use WPE to enable or disable the write posting feature of the EMIF. WPE affects all of the CE spaces. For details about write posting, see section 1.7 on page 1-14.
		0	Disabled
		1	Enabled
6	Reserved		This read-only reserved bit returns 0 when read.
5	MEMCEN		Memory clock enable bit. MEMCEN determines whether the memory clock is enabled at the CLKMEM pin.
		0	Disabled The signal on the CLKMEM pin is held high.
		1	Enabled The memory clock is driven on the CLKMEM pin. The frequency of the clock depends on the MEMFREQ bits.
4	ARDYOFF		ARDY off bit. Use this bit to enable or disable the ARDY signal.
		0	ARDY on The strobe period can be extended with the ARDY signal.
		1	ARDY off The EMIF does not sample the ARDY signal. As a result, the strobe period is never extended past the programmed length.
3	ARDY		ARDY signal status bit. At the time the EMIF is performing an asynchronous read or write operation, the ARDY bit is updated to show the signal level on the asynchronous ready (ARDY) pin. At other times, ARDY is not updated.
		0	ARDY signal is low
		1	ARDY signal is high

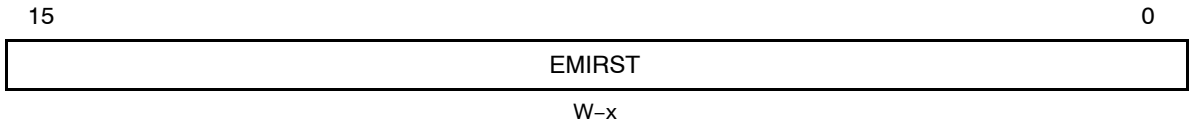
Table 5–2. EMIF Global Control Register (EGCR) Bits (Continued)

Bit	Field	Value	Description
2	HOLD		$\overline{\text{HOLD}}$ signal status bit. The HOLD bit reflects the signal level on the active-low HOLD request ($\overline{\text{HOLD}}$) pin.
		0	$\overline{\text{HOLD}}$ signal is low An external device is driving a HOLD request (a request for the DSP to hold its external memory accesses and allow the external device to access external memory chips). For details about HOLD requests, see section 1.6 on page 1-13.
		1	$\overline{\text{HOLD}}$ signal is high No HOLD request
1	HOLDA		$\overline{\text{HOLDA}}$ signal status bit. The HOLDA bit reflects the signal level on the active-low HOLD acknowledge ($\overline{\text{HOLDA}}$) pin.
		0	$\overline{\text{HOLDA}}$ signal is low The EMIF has acknowledged a HOLD request. The EMIF has relinquished its control over external memory chips, so that an external device can access the chips. Details about HOLD requests are in section 1.6 (page 1-13).
		1	$\overline{\text{HOLDA}}$ signal is high The EMIF is in control of the external memory (it is not in the HOLD state).
0	NOHOLD		$\overline{\text{HOLD}}$ signal disable bit. Use NOHOLD to enable or disable the $\overline{\text{HOLD}}$ pin and thus allow or disallow HOLD requests. Details about HOLD requests are in section 1.6 (page 1-13).
		0	$\overline{\text{HOLD}}$ enabled HOLD requests are accepted by the EMIF.
		1	$\overline{\text{HOLD}}$ disabled No HOLD requests are accepted by the EMIF.

5.3 EMIF Global Reset Register (EMIRST)

Any write to this register (see Figure 5−2 and Table 5−3) resets the logic inside the EMIF. One effect is all pending access requests are cleared. Writing to EMIRST does not change the current configuration values in the EMIF registers. This register cannot be read.

Figure 5−2. EMIF Global Reset Register (EMIRST)



Legend: W = Write; −x = Value after reset is undefined

Table 5−3. EMIF Global Reset Register (EMIRST) Bits

Bit	Field	Value	Description
15−0	EMIRST	0000h−FFFFh	Any write to this register resets the EMIF state machine.

5.4 EMIF Bus Error Status Register (EMIBE)

The bus error status register (EMIBE) is a 16-bit I/O-mapped register. Figure 5–3 and Table 5–4 summarize the fields of EMIBE. This register records the following bus errors that occur during accesses to external asynchronous memory:

- A time-out condition
- An attempt to write to a CE space that is configured for 8-bit-wide asynchronous memory. Such writes are not supported by the EMIF.

For each bus error recognized by the EMIF:

- The EMIF terminates the external transaction. The EMIF advances to the hold period and completes the programmed number of hold cycles.
- If the request came from the DMA controller, all pending DMA requests are flushed from the EMIF.
- The EMIF sets at least two bits in EMIBE:
  - One of the CE bits (bits 10 through 7), to identify which CE space was being accessed when the error occurred.
  - One of the requester bits (6 through 2 and 0), to identify which DSP resource requested the external memory access.
- If the error was a time-out condition, the EMIF also sets the TIME bit (bit 12).

As explained in section 1.3 (page 1-9), there is a difference between dual data accesses and long data accesses. This difference is reflected in the way bus errors are recorded in EMIBE:

- A long data write uses two data buses (EB and FB) but is a single E-bus request. If a bus error occurs when the EMIF attempts to service this request, the EBUS bit is set, but the FBUS bit is not.
- A long data read uses data buses CB and DB but is a single D-bus request. If a bus error occurs during servicing, the DBUS bit is set, but the CBUS bit is not.
- For any dual data access, the EMIF receives separate requests from two CPU buses. The error bits corresponding to either or both of the buses may be set.

After EMIBE is read, it is automatically cleared. If a second bus error occurs before the first error is read and cleared, the content of EMIBE is a combination of the results from the two failed accesses. Consider the following example: A D-bus request to the CE2 space results in a bus error. Before EMIBE is read, an E-bus request to the CE1 space results in a second bus error. When EMIBE is read, bits CE2, CE1, EBUS, and DBUS will all be 1.

EMIF bus errors have interrupt activity associated with them. If the requester was a CPU bus, the EMIF sends a bus-error interrupt request to the CPU. If the requester was the DMA controller and a time-out error has occurred, the EMIF sends a time-out signal to the DMA controller. The DMA controller can ignore the signal or send a bus-error interrupt request to the CPU. The bus-error interrupt is maskable; the CPU ignores it or services it depending on whether the interrupt is properly enabled.

Figure 5–3. EMIF Bus Error Status Register (EMIBE)

15	14	13	12	11	10	9	8
Reserved†			TIME	Reserved†	CE3	CE2	CE1
R–0			R–0	R–0	R–0	R–0	R–0
7	6	5	4	3	2	1	0
CE0	DMA	FBUS	EBUS	DBUS	CBUS	Reserved	PBUS
R–0	R–0	R–0	R–0	R–0	R–0	R–0	R–0

Legend: R = Read; -*n* = Value after reset
† The read states of these reserved bits are undefined.

Table 5–4. EMIF Bus Error Status Register (EMIBE) Bits

Bit	Field	Value	Description
15–13	Reserved		These are read-only reserved bits. The read states of these bits are undefined.
12	TIME		Time-out error status bit. The EMIF sets TIME when a time-out error occurs during an access to asynchronous memory.
		0	No error
		1	Error
11	Reserved		This is a read-only reserved bit. The read state of this bit is undefined.

Table 5–4. EMIF Bus Error Status Register (EMIBE) Bits (Continued)

Bit	Field	Value	Description
10	CE3		CE3 error status bit. The EMIF sets CE3 when an error occurs during an access to memory in the address range defined as the CE3 space.
		0	No error
		1	Error
9	CE2		CE2 error status bit. The EMIF sets CE2 when an error occurs during an access to memory in the address range defined as the CE2 space.
		0	No error
		1	Error
8	CE1		CE1 error status bit. The EMIF sets CE1 when an error occurs during an access to memory in the address range defined as the CE1 space.
		0	No error
		1	Error
7	CE0		CE0 error status bit. The EMIF sets CE0 when an error occurs during an access to memory in the address range defined as the CE0 space.
		0	No error
		1	Error
6	DMA		DMA error status bit. The EMIF sets DMA when an error occurs during an access requested by the DMA controller.
		0	No error
		1	Error
5	FBUS		F-bus error status bit. The EMIF sets FBUS when an error occurs while the EMIF is servicing an F-bus request from the CPU. An F-bus request is for a 16-bit data write.
		0	No error
		1	Error

Table 5–4. EMIF Bus Error Status Register (EMIBE) Bits (Continued)

Bit	Field	Value	Description
4	EBUS		E-bus error status bit. The EMIF sets EBUS when an error occurs while the EMIF is servicing an E-bus request from the CPU. An E-bus request is for a 16-bit or 32-bit data write.
		0	No error
		1	Error
3	DBUS		D-bus error status bit. The EMIF sets DBUS when an error occurs while the EMIF is servicing a D-bus request from the CPU. A D-bus request is for a 16-bit or 32-bit data read.
		0	No error
		1	Error
2	CBUS		C-bus error status bit. The EMIF sets CBUS when an error occurs while the EMIF is servicing a C-bus request from the CPU. A C-bus request is for a 16-bit data read.
		0	No error
		1	Error
1	Reserved		This read-only reserved bit returns 0 when read.
0	PBUS		P-bus error status bit. The EMIF sets PBUS when an error occurs while the EMIF is servicing a P-bus request from the CPU or from the instruction cache. A P-bus request is for a 32-bit instruction fetch.
		0	No error
		1	Error

5.5 CE Space Control Registers (CEn1, CEn2, and CEn3 in Each CE Space)

The external memory map is divided into CE spaces (see section 1.4 on page 1-11). Each CE space has three CE space control registers of the form shown in Figure 5−4. The contents of these registers are described in Table 5−6 through Table 5−8. These are 16-bit I/O-mapped registers used primarily for configuring accesses to asynchronous memory.

With the MTYPE bit, select the memory type for the given CE space. If you choose an asynchronous memory type, use the other bits in the CE space control registers to define the timing parameters. For details about the asynchronous timing parameters, see section 2.2 on page 2-3. If you choose a synchronous memory type, the EMIF ignores all of the bits but MTYPE.

Notes:

1) The SETUP and STROBE fields have a minimum count of 1, and because of this, a 0 in one of these fields is interpreted by the DSP as a 1. For the first access (even if there is only one), the setup period has a minimum of 2 cycles. Table 5−5 shows how the setup constraints affect the number of cycles in the first and subsequent setup periods.

2) Writing 0, 1, or 2 to a SETUP field results in a programmed setup period of 1 cycle. Writing a number n larger than 2 results in a programmed setup period of n cycles. This behavior is also reflected in Table 5−5.

Table 5−5. Setup Period Cycles For First and Subsequent Accesses

SETUP Bits	Setup Period For First Access (CPU Clock Cycles)	Setup Period For Subsequent Accesses (CPU Clock Cycles)
0	2	1
1	2	1
2	2	1
$3 \leq n \leq 15$	n	n

Figure 5–4. CE Space Control Registers (CEn1, CEn2, and CEn3) for Each CE Space

CEn1

15	14	12	11	8
Reserved	MTYPE		RDSETUP	
R–0	R/W–010		R/W–1111	

7	2	1	0
RDSTROBE			RDHOLD
R/W–11 1111			R/W–11

CEn2

15	14	13	12	11	8
RDEXHLD		WREXHLD		WRSETUP	
R/W–01		R/W–01		R/W–1111	

7	2	1	0
WRSTROBE			WRHOLD
R/W–11 1111			R/W–11

CEn3

15	8
Reserved	
R–0	

7	0
TIMOUT	
R/W–0	

Legend: R = Read; W = Write; *-n* = Value after reset

Table 5–6. CE Space Control Register 1 (CEn1) Bits

Bit	Field	Value	Description
15	Reserved		This read-only reserved bit returns 0 when read.
14–12	MTYPE		Memory type bits. Each of the CE spaces has MTYPE bits. For each CE space, use MTYPE to select one of the following memory types:
		000b†	Asynchronous, 8-bit data bus width
		001b	Asynchronous, 16-bit data bus width
		010b	Asynchronous, 32-bit data bus width
		011b	Synchronous DRAM (SDRAM), 16-bit or 32-bit data bus width
		100b	Synchronous burst SRAM (SBSRAM), 32-bit data bus width
		other	Reserved (do not use)
11–8	RDSETUP		Read setup period bits. For each CE space that contains asynchronous memory, load the associated RDSETUP bits with a setup period for read operations in that CE space.
		0, 1, or 2	1 CPU clock cycle
		$3 \leq n \leq 15$	n CPU clock cycles
7–2	RDSTROBE		Read strobe period bits. For each CE space that contains asynchronous memory, load the associated RDSTROBE bits with a strobe period for read operations in that CE space.
		0 or 1	1 CPU clock cycle
		$2 \leq n \leq 63$	n CPU clock cycles
1–0	RDHOLD		Read hold period bits. For each CE space that contains asynchronous memory, load the associated RDHOLD bits with a hold period for read operations in that CE space. **Note:** If the ARDY pin extends the strobe period, the EMIF does not allow a hold period of 0 cycles. If RDHOLD = 0 and ARDY is used, the EMIF automatically generates a 1-cycle hold period.
		$0 \leq n \leq 3$	n CPU clock cycles

† When MTYPE = 000b, the EMIF can only read from external memory; writes to 8-bit-wide memory are not possible.

Table 5−7. CE Space Control Register 2 (CEn2) Bits

Bit	Field	Value	Description
15−14	RDEXHLD		Read extended hold period bits. For each CE space that contains asynchronous memory, load the associated RDEXHLD bits with an extended hold period for read operations in that CE space. The EMIF automatically adds 1 to the value programmed.
		$0 \le n \le 3$	(n + 1) CPU clock cycles
13−12	WREXHLD		Write extended hold period bits. For each CE space that contains asynchronous memory, load the associated WREXHLD bits with an extended hold period for write operations in that CE space. The EMIF automatically adds 1 to the value programmed.
		$0 \le n \le 3$	(n + 1) CPU clock cycles
11−8	WRSETUP		Write setup period bits. For each CE space that contains asynchronous memory, load the associated WRSETUP bits with a setup period for write operations in that CE space.
		0, 1, or 2	1 CPU clock cycle
		$3 \le n \le 15$	n CPU clock cycles
7−2	WRSTROBE		Write strobe period bits. For each CE space that contains asynchronous memory, load the associated WRSTROBE bits with a strobe period for write operations in that CE space.
		0 or 1	1 CPU clock cycle
		$2 \le n \le 63$	n CPU clock cycles
1−0	WRHOLD		Write hold period bits. For each CE space that contains asynchronous memory, load the associated WRHOLD bits with a hold period for write operations in that CE space. **Note:** If the ARDY pin extends the strobe period, the EMIF does not allow a hold period of 0 cycles. If WRHOLD = 0 and ARDY is used, the EMIF automatically generates a 1-cycle hold period.
		$0 \le n \le 3$	n CPU clock cycles

Table 5–8. CE Space Control Register 3 (CEn3) Bits

Bit	Field	Value	Description
15–8	Reserved		These read-only reserved bits return 0s when read.
7–0	TIMOUT		Time-out bits. For each CE space that contains asynchronous memory, load the associated TIMOUT bits with a time-out value (n) for all operations in that CE space, or disable the time-out feature by clearing TIMOUT.
		0	Time-out feature disabled (default after reset)
		$1 \leq n \leq 255$	An internal counter counts the number of cycles that the asynchronous ready signal (ARDY) is sampled low (indicating that the memory is not ready for an access). If ARDY is sampled low for n CPU clock cycles, the EMIF signals a time-out error.

5.6 SDRAM Control Registers (SDC1 and SDC2)

The SDRAM control registers control SDRAM parameters for all CE spaces that specify an SDRAM memory type in the MTYPE field of its associated CE space control register 1. Each CE space with SDRAM should be compatible with the same refresh, timing, and page characteristics. The delay values in the control registers are in CLKMEM cycles. The control register fields are shown in Figure 5−5. Table 5−9 and Table 5−10 describe the timing parameters that must be configured with the control registers. The nonconfigurable SDRAM timing parameters are listed in Table 5−11.

Figure 5−5. SDRAM Control Registers (SDC1 and SDC2)

SDC1

15	11	10	9	8
TRC		SDSIZE	SDWID	RFEN
R/W−11111		R/W−0	R/W−0	R/W−1

7	4	3	0
TRCD		TRP	
R/W-0100		R/W-1000	

SDC2

15	11	10	9	8
Reserved†		SDACC	TMRD	
R/W−0		R/W−0	R/W−11	

7	4	3	0
TRAS		TACTV2ACTV	
R/W−1111		R/W−1111	

Legend: R = Read; W = Write; *-n* = Value after reset
† Always write 0s to these reserved bits.

Table 5–9. SDRAM Control Register 1 (SDC1) Bits

Bit	Field	Value	Description
15–11	TRC	0–31	TRC specifies the tRC ($\overline{\text{SDRAS}}$ Cycle Time) value of the SDRAM in CLKMEM cycles from REFR to REFR/MRS/ACTV command. TRC = (tRC/CLKMEM) – 1
10–9	SDSIZE:SDWID		SDSIZE selects a memory size of 64M bits (SDSIZE = 0) or 128M bits (SDSIZE = 1). SDWID selects a memory width of 16 bits (SDWID = 0) or 32 bits (SDWID = 1). Together, these bits define what type of SDRAM chips are to be used with the EMIF. This setting applies to all CE spaces that are configured for SDRAM.
		00b	4M x 16 bits (64M bits)
		01b	2M x 32 bits (64M bits)
		10b	8M x 16 bits (128M bits)
		11b	4M x 32 bits (128M bits)
8	RFEN		RFEN enables or disables the capability of the EMIF to send SDRAM refresh commands.
		0	SDRAM refresh disabled
		1	SDRAM refresh enabled
7–4	TRCD	0–15	TRCD specifies the tRCD (Activate to Command or $\overline{\text{SDRAS}}$ to $\overline{\text{SDCAS}}$ Delay) value of the SDRAM in CLKMEM cycles from ACTV to READ/WRITE command. TRCD = (tRCD/CLKMEM) – 1
3–0	TRP	0–15	TRP specifies the tRP ($\overline{\text{SDRAS}}$ Precharge Time) value of the SDRAM in CLKMEM cycles from DCAB to REFR/ACTV/MRS command. TRP = (tRP/CLKMEM) – 1

Table 5−10. SDRAM Control Register 2 (SDC2) Bits

Bit	Field	Value	Description
15−11	Reserved	0	Always write 0s to these reserved bits.
10	SDACC		SDACC selects an EMIF data bus width.
		0	The SDRAM data bus interface is 16 bits wide. The EMIF uses pins D[15:0] for data.
		1	The SDRAM data bus interface is 32 bits wide. The EMIF uses pins D[31:0] for data. Use this setting for a 32-bit-wide memory chip, or if two 16-bit-wide memory chips are used.
9−8	TMRD	0−3	TMRD specifies the tMRD (Mode Register set to ACTV/DCAB/REFR Delay) value in CLKMEM clock cycles. TMRD = (tMRD/CLKMEM) − 1
7−4	TRAS	0−15	TRAS specifies the tRAS ($\overline{\text{SDRAS}}$ Active Time) value in CLKMEM clock cycles. TRAS = (tRAS/CLKMEM) − 1
3−0	TACTV2ACTV	0−15	TACTV2ACTV specifies the tRRD ($\overline{\text{SDRAS}}$ to $\overline{\text{SDRAS}}$ Bank Activate Delay) value in CLKMEM clock cycles. TACTV2ACTV = (tRRD/CLKMEM) − 1

Table 5−11. Nonconfigurable SDRAM Timing Parameters

Parameter	Description	CLKMEM Cycles
tCL	CAS latency	3
tRD2DCAB	Delay from a READ command to a DCAB command	4
tWR2DCAB	Delay from a WRT command to a DCAB command	4
tWR2WR	Delay from a WRT command to another WRT command	1
tRD2RD	Delay from a READ command to another READ command	1
tRD2WR	Delay from a READ command to a WRT command	5
tWR2RD	Delay from a WRT command to a READ command	5

5.7 SDRAM Period and Counter Registers (SDPER and SDCNT)

The SDRAM period register (SDPER) and the SDRAM counter register (SDCNT) are summarized in Figure 5–6 and the tables following the figure. SDPER is used to set the period between trickle refreshes in units of CLKMEM cycles. For the minimum refresh period required, consult the SDRAM manufacturer's data sheet.

When the refresh enable bit is set (RFEN = 1 in SDC1), a 16-bit refresh period counter is loaded from SDPER, and the counter is decremented every 1 CLKMEM cycle. When the counter reaches 0:

- The counter is reloaded from SDPER.
- If SDRAM is not being accessed at the time, a REFR command is sent to all SDRAM chips.
- If SDRAM is being accessed at the time, the trickle refresh request is recorded so that it can be serviced later. If three requests have been received but not serviced, an urgent refresh request is generated, so that the SDRAM is refreshed immediately.

For more details about trickle refresh and urgent refresh requests, see section 4.9.1 on page 4-21.

Figure 5–6. SDRAM Period Register (SDPER) and SDRAM Counter Register (SDCNT)

SDPER

15	12	11	0
Reserved		PERIOD	
R–0		R/W–080h	

SDCNT

15	12	11	0
Reserved		COUNTER	
R–0		R–080h	

Legend: R = Read; W = Write; *-n* = Value after reset

Table 5−12. SDRAM Period Register (SDPER) Bits

Bit	Field	Value	Description
15−12	Reserved	0	These read-only reserved bits always return 0s.
11−0	PERIOD	0−4095	This field contains the refresh period in CLKMEM cycles. The default after reset is 128 CLKMEM cycles (080h).

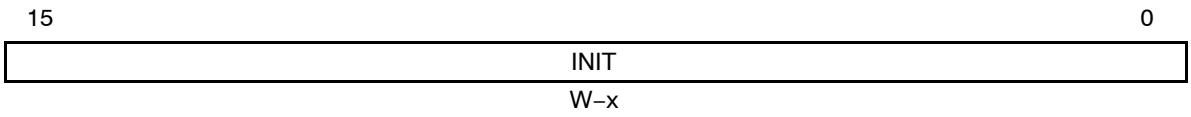
Table 5−13. SDRAM Counter Register (SDCNT) Bits

Bit	Field	Value	Description
15−12	Reserved	0	These read-only reserved bits always return 0s.
11−0	COUNTER	0−4095	This field contains the current value of the refresh period counter. The default starting value after reset is 128 CLKMEM cycles (080h).

5.8 SDRAM Initialization Register (INIT)

INIT is summarized by Figure 5−7 and Table 5−14. Any write to this register will cause an SDRAM initialization sequence within each CE space configured for SDRAM. After a hardware reset or powering up the C55x device, a write to this register should be performed following configuration of all CE spaces, and prior to accessing SDRAM. For details on the initialization sequence, see section 4.7 on page 4-17.

Figure 5−7. SDRAM Initialization Register (INIT)



Legend: R = Read; W = Write; -x = Value is not defined after reset

Table 5−14. SDRAM Initialization Register (INIT) Bits

Bit	Field	Value	Description
15−0	INIT	0000h−FFFFh	Write any value to this register to cause an SDRAM initialization sequence. The write should be performed following configuration of all CE spaces, and prior to accessing SDRAM.

This page is intentionally left blank.

Appendix A

Details of Instruction Fetches and Data Accesses

This appendix contains the details of how the EMIF communicates with external memory for different memory widths and data sizes.

Topic **Page**

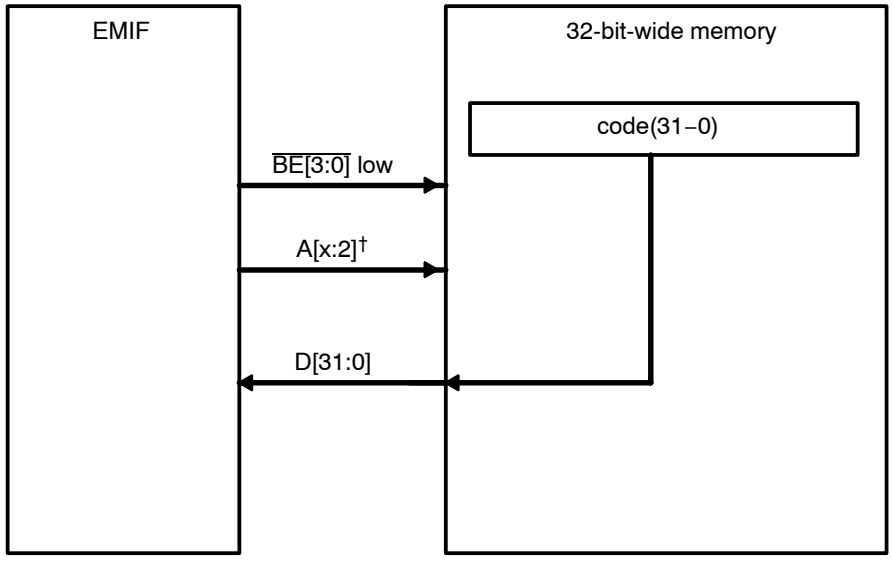
A.1 Instruction Fetches via the EMIF

When fetching instruction code from external memory, the CPU or the instruction cache sends an access request to the EMIF. The EMIF reads 32 bits from the external memory and then passes all 32 bits to the CPU or the instruction cache. The EMIF can perform a 32-bit instruction fetch from external memory that is 32, 16, or 8 bits wide.

A.1.1 Instruction Fetch From 32-Bit-Wide Memory

Figure A–1 shows how the EMIF behaves when reading instruction code from 32-bit-wide external memory. The least significant address line required by 32-bit-wide memory is A2. The whole external data bus, D[31:0], is used to transport the data from external memory to the DSP. During the 32-bit access, the EMIF drives low all four of the byte enable signals, $\overline{\text{BE}}[3:0]$. After the access, the EMIF passes all 32 bits to the requester (the CPU or the instruction cache).

Figure A–1. Instruction Fetch From 32-Bit-Wide External Memory



† The least significant address line required is A2. The type of memory accessed determines which other address lines are required.

A.1.2 Instruction Fetch From 16-Bit-Wide Memory

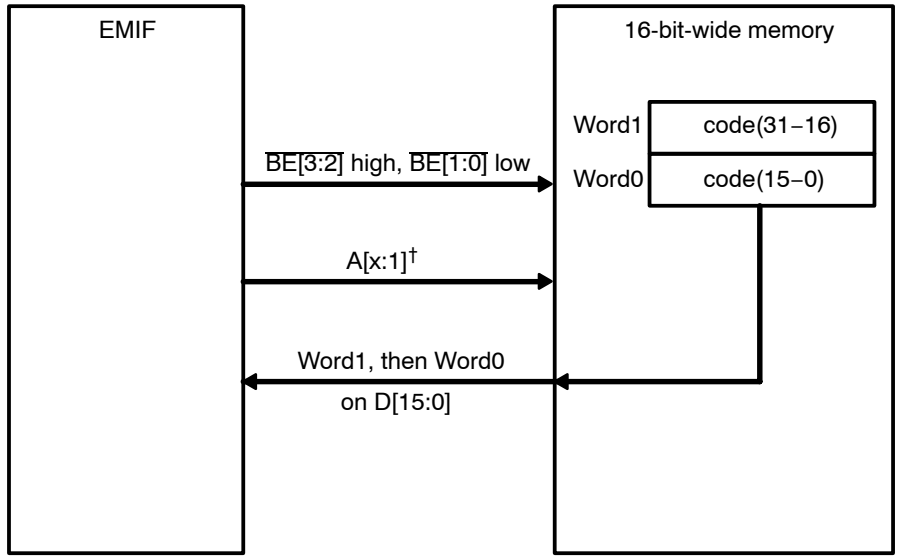
Figure A–2 illustrates the data transfers involved in an instruction fetch from 16-bit-wide external memory. The EMIF places a word address on the address lines. The 32-bit access is performed as two consecutive 16-bit transfers. For the second transfer, the EMIF automatically increments the first address by 1 to create the second address.

For both 16-bit transfers, the EMIF uses data lines D[15:0]. The 32-bit code block is transferred in the following manner:

1) Bits 31 through 16 of the code block are read at the first address.
2) Bits 15 through 0 are read at the second address.

During both transfers, $\overline{BE3}$ and $\overline{BE2}$ are driven high (inactive), and $\overline{BE1}$ and $\overline{BE0}$ are driven low. The EMIF packs the two words internally and then passes all 32 bits of code to the requester (the CPU or the instruction cache).

Figure A–2. Instruction Fetch From 16-Bit-Wide External Memory



† The least significant address line required is A1. The type of memory accessed determines which other address lines are required.

A.1.3 Instruction Fetch From 8-Bit-Wide Memory

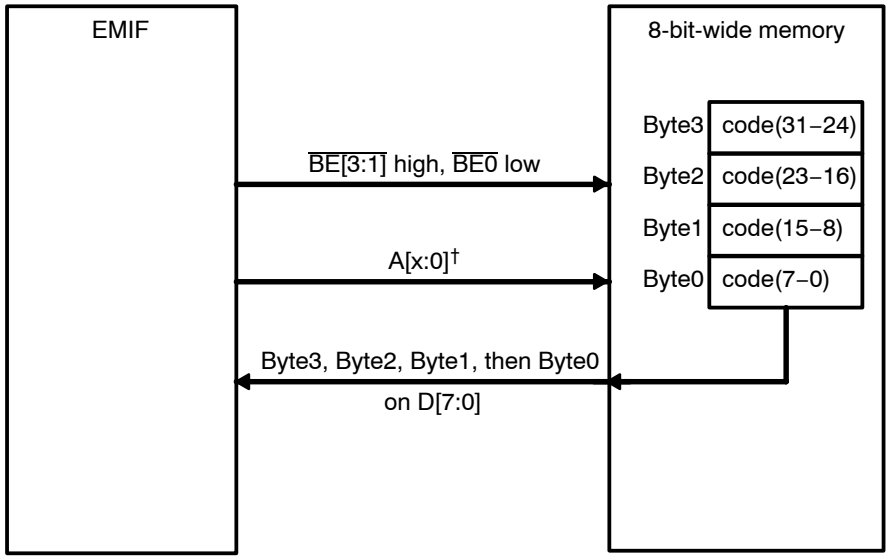
As shown in Figure A–3, when the EMIF performs an instruction fetch from 8-bit-wide memory, the EMIF places a byte address on the address lines. The 32-bit access is performed as four consecutive 8-bit transfers. For the second, third, and fourth transfers, the EMIF automatically generates a new address by incrementing the previous address by 1.

For all four 8-bit transfers, the EMIF uses data lines D[7:0]. As shown in the figure, the 32-bit code block is transferred in the following manner:

1) Bits 31 through 24 of the code block are read at the first address.
2) Bits 23 through 16 are read at the second address.
3) Bits 15 through 8 are read at the third address.
4) Bits 7 through 0 are read at the fourth address.

During these transfers, $\overline{BE3}$, $\overline{BE2}$, and $\overline{BE1}$ are driven high (inactive), and $\overline{BE0}$ is driven low. The EMIF packs the four bytes internally and then passes all 32 bits of code to the requester (the CPU or the instruction cache).

Figure A–3. Instruction Fetch From 8-Bit-Wide External Memory



† The least significant address line required is A0. The type of memory accessed determines which other address lines are required.

A.2 32-Bit Data Accesses of External Memory

A 32-bit data access is generated by a CPU instruction or a DMA controller operation that reads or writes a 32-bit value. For each 32-bit read or write operation initiated by the CPU, the data is carried on two 16-bit CPU buses. For read operations, the C and D buses carry the data from the EMIF to the CPU. For write operations, the E and F buses carry the data from the CPU to the EMIF.

The EMIF can read 32-bit data from external memory that is 32,16, or 8 bits wide. When the EMIF writes 32-bit data to external memory, the memory must be 32 or 16 bits wide; the EMIF cannot write to 8-bit-wide memory.

A.2.1 32-Bit Data Access of 32-Bit-Wide Memory

The two parts of Figure A–4 illustrate the data transfers involved when the EMIF accesses 32-bit data in 32-bit-wide external memory. The least significant address line required by 32-bit-wide memory is A2. The whole external data bus, D[31:0], is used to transport the data between the DSP and the external memory. All four of the byte enable signals, $\overline{\text{BE}}$[3:0], are forced low (active).

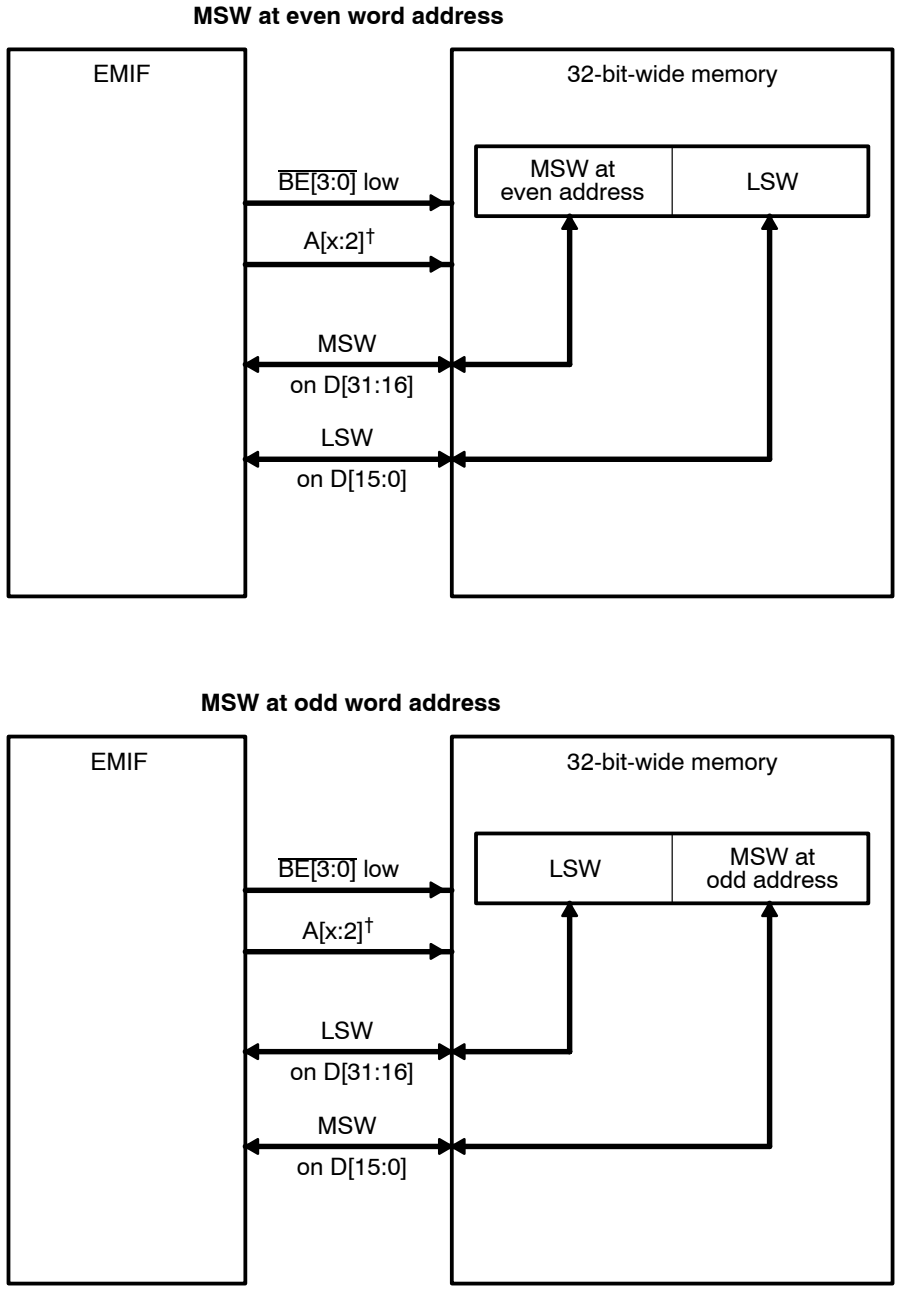
The external memory chip needs only bits 21–2 of the internal address. However, if the CPU has requested the access, the EMIF uses bit 1 to determine the positions of the most significant word (MSW) and least significant word (LSW) of the 32-bit value (see Table A–1). The relative positions of the MSW and LSW determine how the EMIF data lines are used. D[31:16] carry the word to/from the even word address. Lines D[15:0] carry the word to/from the odd word address.

If the DMA controller is requesting a 32-bit data access, the MSW must be at an even address.

Table A–1. The Role of Internal Address Bit 1 During a 32-Bit Data Access of 32-Bit-Wide External Memory

Internal Address Bit 1	MSW and LSW Positions	Use of D[31:0]
0	The MSW is at an even word address, and the LSW is at the following odd word address.	D[31:16] carry the MSW. D[15:0] carry the LSW.
1	The MSW is at an odd word address, and the LSW is at the previous even word address.	D[31:16] carry the LSW. D[15:0] carry the MSW.

Figure A–4. 32-Bit Data Access of 32-Bit-Wide External Memory



† The least significant address line required is A2. The type of memory accessed determines which other address lines are required.

A.2.2 32-Bit Data Access of 16-Bit-Wide Memory

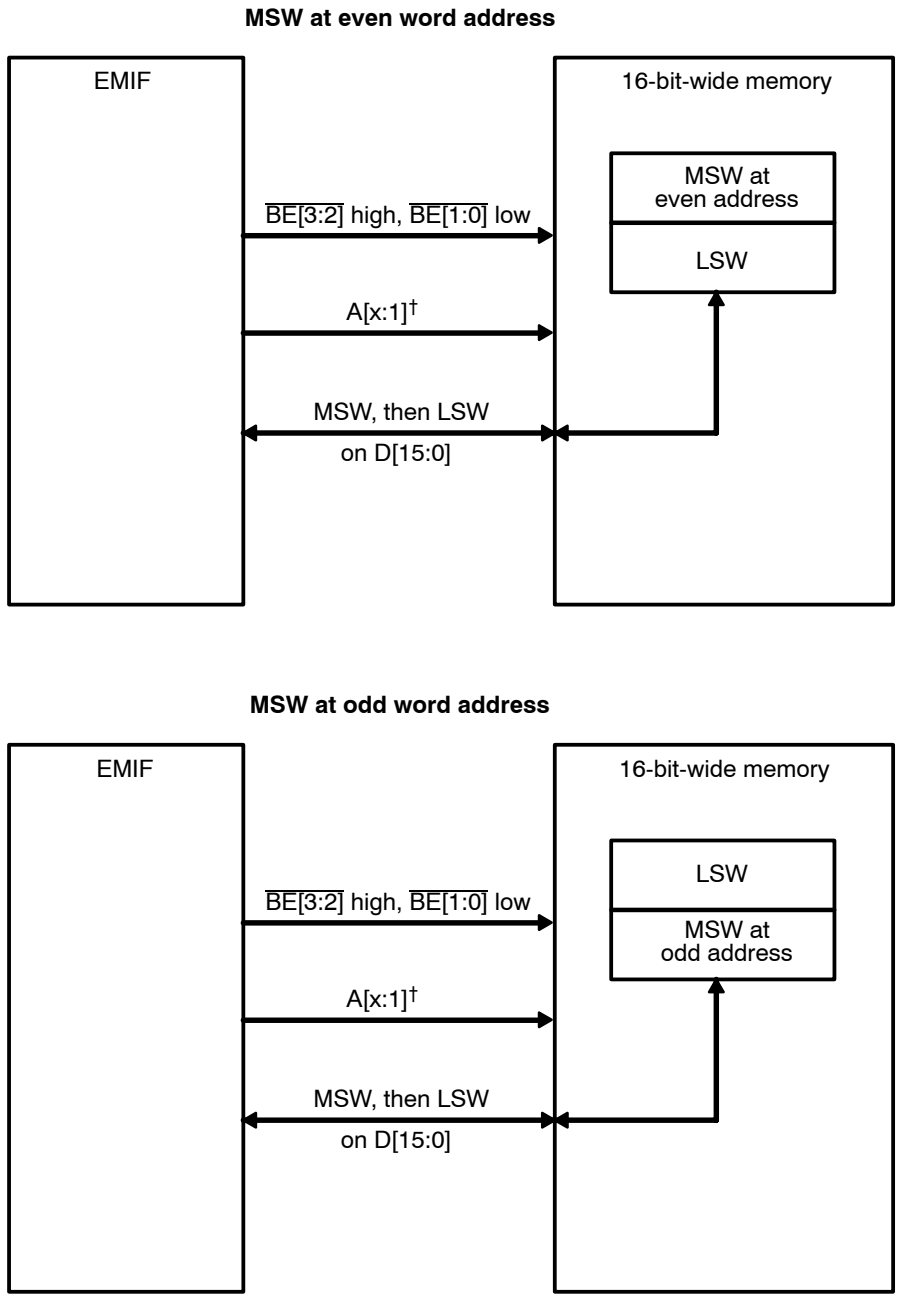
The two parts of Figure A–5 illustrate the data transfers involved in a 32-bit data access of 16-bit-wide external memory. The least significant address line required by 16-bit-wide memory is A1. Data bus lines D[15:0] are used to transport the data between the DSP and the external memory. During an access, $\overline{BE3}$ and $\overline{BE2}$ are driven high (inactive), and $\overline{BE1}$ and $\overline{BE0}$ are driven low.

The 32-bit access is performed as two consecutive 16-bit transfers across D[15:0]. If the CPU has requested the access, the EMIF automatically generates the second address based on the position of the MSW (see Table A–2). If the DMA controller is requesting a 32-bit data access, the MSW must be at an even address.

Table A–2. The Role of Internal Address Bit 1 During a 32-Bit Data Access of 16-Bit-Wide External Memory

Internal Address Bit 1	MSW and LSW Positions	Generation of LSW Address
0	The MSW is at an even word address, and the LSW is at the following odd word address.	After the first access, the EMIF adds 1 to the MSW address to generate the LSW address.
1	The MSW is at an odd word address, and the LSW is at the previous even word address.	After the first access, the EMIF subtracts 1 from the MSW address to generate the LSW address.

Figure A–5. 32-Bit Data Access of 16-Bit-Wide External Memory



† The least significant address line required is A1. The type of memory accessed determines which other address lines are required.

A.2.3 32-Bit Data Read From 8-Bit-Wide Memory

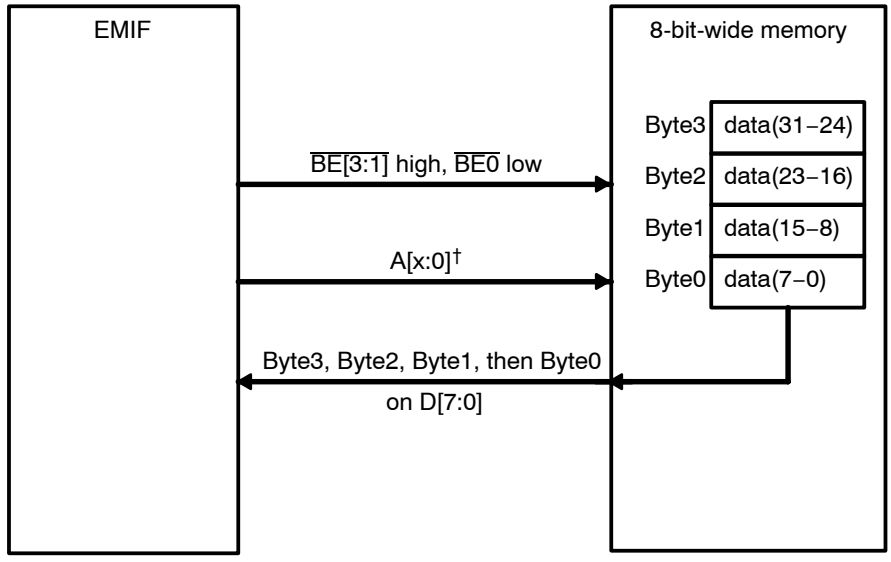
When the EMIF reads a 32-bit value from 8-bit-wide memory, the EMIF places a byte address on the address lines. As shown in Figure A–6, the 32-bit access is performed as four consecutive 8-bit transfers. For the second, third, and fourth transfers, the EMIF automatically generates a new address by incrementing the previous address by 1.

For all four 8-bit transfers, the EMIF uses data lines D[7:0]. As shown in the figure, the 32-bit value is transferred in the following manner:

1) Bits 31 through 24 of the value are read at the first address.
2) Bits 23 through 16 are read at the second address.
3) Bits 15 through 8 are read at the third address.
4) Bits 7 through 0 are read at the fourth address.

During these transfers, $\overline{BE3}$, $\overline{BE2}$, and $\overline{BE1}$ are driven high (inactive), and $\overline{BE0}$ is driven low. The EMIF packs the four bytes internally and then passes all 32 bits of data to the requester (the CPU or the DMA controller).

Figure A–6. 32-Bit Data Read From 8-Bit-Wide External Memory



† The least significant address line required is A0. The type of memory accessed determines which other address lines are required.

A.3 16-Bit Data Accesses of External Memory

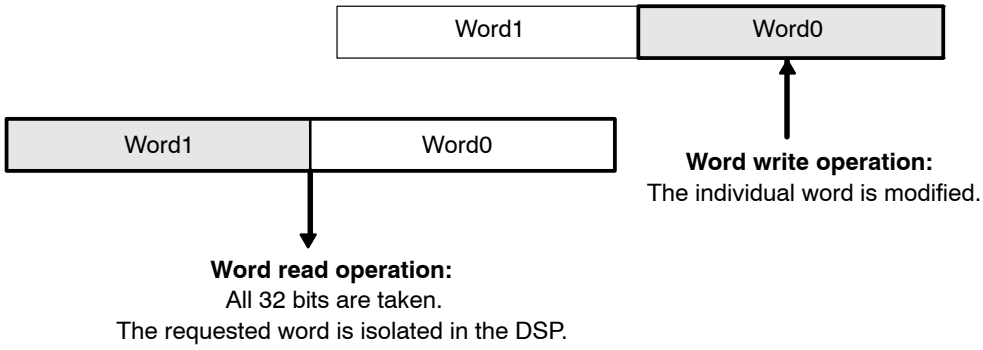
A 16-bit data access is generated by a CPU instruction or a DMA controller operation that reads or writes a 16-bit value. For CPU operations, the D bus carries read data, and the E bus carries write data.

The EMIF can read 16-bit data from external memory that is 32,16, or 8 bits wide. When the EMIF writes 16-bit data to external memory, the memory must be 32 or 16 bits wide; the EMIF cannot write to 8-bit-wide memory.

A.3.1 16-Bit Data Access of 32-Bit-Wide Memory

As shown in Figure A–7, when the EMIF makes a 16-bit access in 32-bit-wide external memory, the actual width of the access is different for a read operation and a write operation. When a word is written to external memory, the EMIF modifies an individual word. However, when a word is read from external memory, the EMIF reads the full width of the memory, and the requested word is isolated in the DSP.

Figure A–7. Accessing 16-Bit Data in 32-Bit-Wide Memory

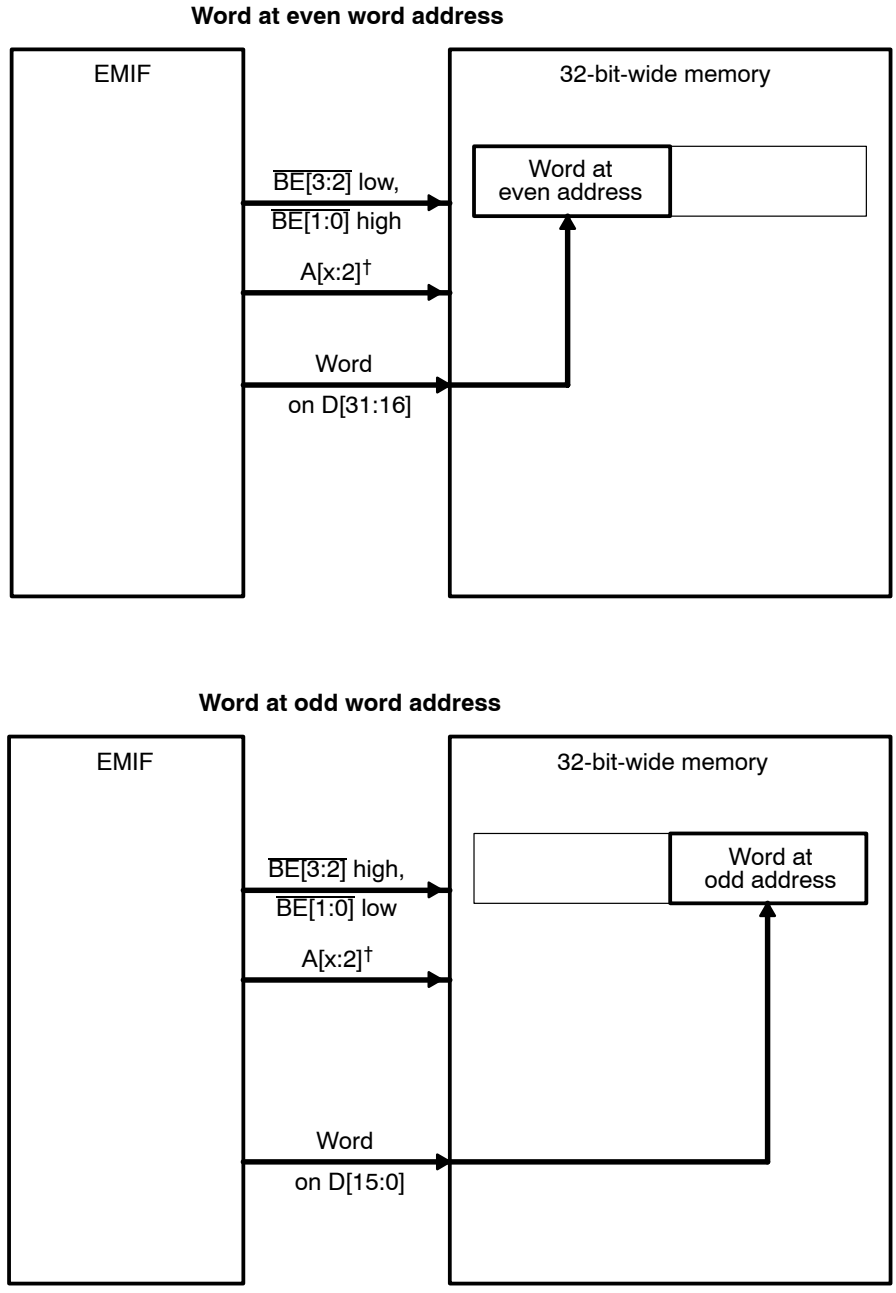


For a read operation, the EMIF uses the full data bus, D[31:0], and drives all four byte enable signals low (active). For a write operation, the EMIF uses bit 1 of the internal address to determine which half of the data bus is used and which byte enable signals are active (see Table A–3 and Figure A–8).

Table A–3. The Role of Internal Address Bit 1 When the EMIF Writes 16-Bit Data to 32-Bit-Wide External Memory

Internal Address Bit 1	Word Written To ...	Data Lines Used	Byte Enable Signal Levels
0	Even word address	D[31:16]	$\overline{BE[3:2]}$ low (active) $\overline{BE[1:0]}$ high
1	Odd word address	D[15:0]	$\overline{BE[3:2]}$ high $\overline{BE[1:0]}$ low (active)

Figure A–8. Writing 16-Bit Data to 32-Bit-Wide External Memory

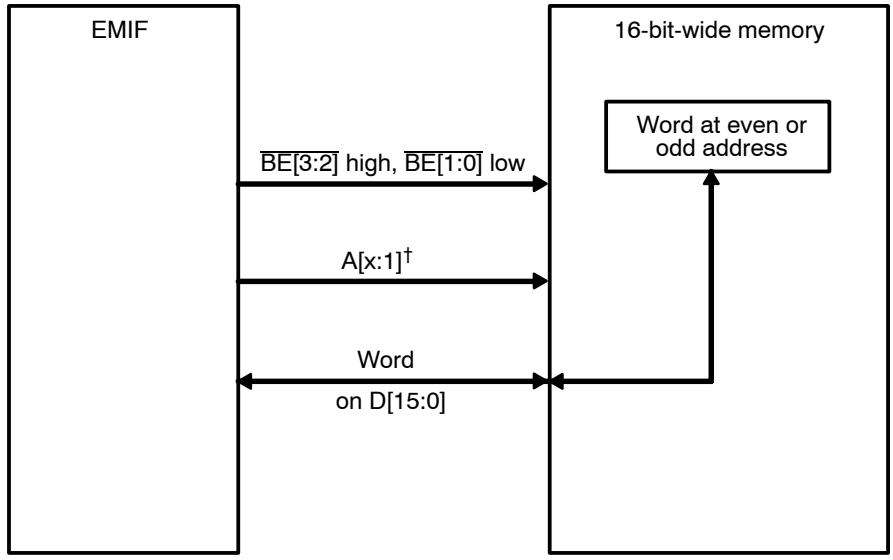


† The least significant address line required is A2. The type of memory accessed determines which other address lines are required.

A.3.2 16-Bit Data Access of 16-Bit-Wide Memory

Figure A–9 illustrates the data transfer involved when the EMIF makes a 16-bit data access in external memory that is 16 bits wide. The least significant address line required by 16-bit-wide memory is A1. Data bus lines D[15:0] are used to transport the data between the DSP and the external memory. During an access, the EMIF drives $\overline{BE3}$ and $\overline{BE2}$ high and drives $\overline{BE1}$ and $\overline{BE0}$ low.

Figure A–9. 16-Bit Data Access of 16-Bit-Wide External Memory



† The least significant address line required is A1. The type of memory accessed determines which other address lines are required.

A.3.3 16-Bit Data Read From 8-Bit-Wide Memory

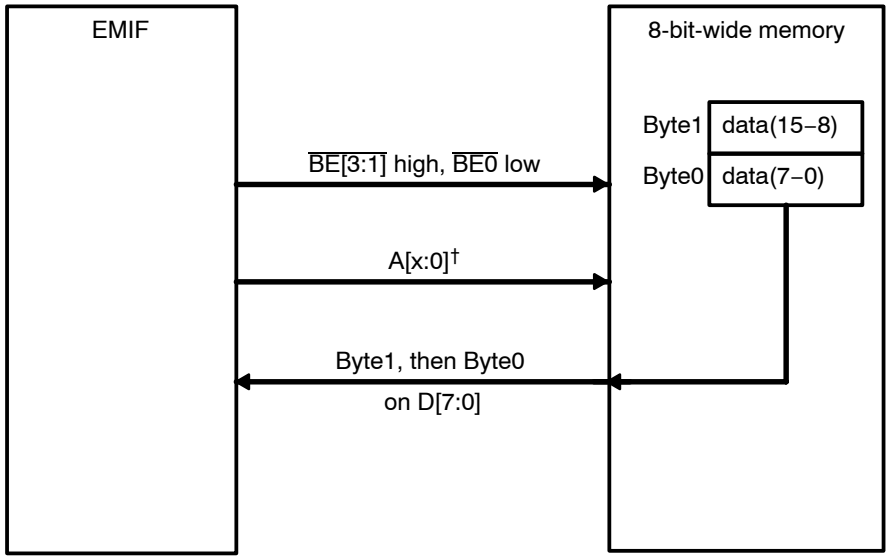
When the EMIF reads a 16-bit value from 8-bit-wide memory, the EMIF places a word address on the address lines. As shown in Figure A–10, the 16-bit access is performed as two consecutive 8-bit transfers. For the second transfer, the EMIF automatically generates a new address by incrementing the previous address by 1.

For both 8-bit transfers, the EMIF uses data lines D[7:0]. As shown in the figure, the 16-bit value is transferred in the following manner:

1) Bits 15 through 8 of the value are read at the first address.
2) Bits 7 through 0 are read at the second address.

While the data is transferred, the EMIF drives $\overline{BE3}$, $\overline{BE2}$, and $\overline{BE1}$ high and drives $\overline{BE0}$ low. The EMIF packs the two bytes internally and then passes all 16 bits of data to the requester.

Figure A−10. 16-Bit Data Read From 8-Bit-Wide External Memory



† The least significant address line required is A0. The type of memory accessed determines which other address lines are required.

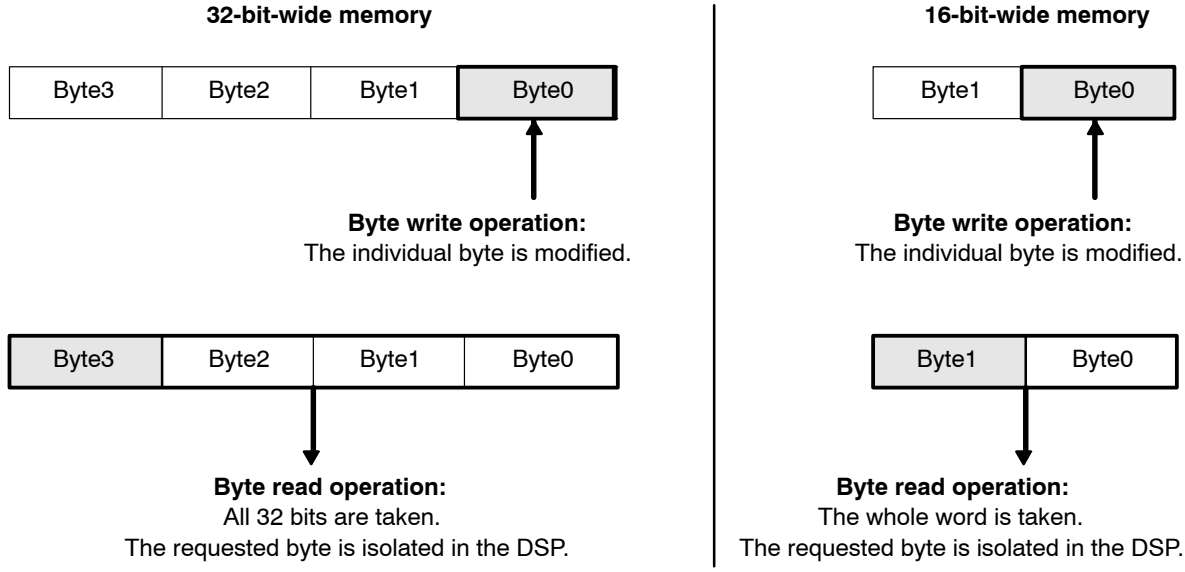
A.4 8-Bit Data Accesses of External Memory

Some CPU instructions and DMA controller operations access 8-bit data (bytes). The CPU buses used are the D bus (for byte read operations) and the E bus (for byte write operations).

The EMIF can read 8-bit data from external memory that is 32,16, or 8 bits wide. When the EMIF writes 8-bit data to external memory, the memory must be 32 or 16 bits wide; the EMIF cannot write to 8-bit-wide memory.

As shown in Figure A–11, when the memory is wider than 8 bits, the actual width of the memory access is different for byte read operations and byte write operations. When a byte is written to external memory, the EMIF modifies an individual byte. However, when the EMIF reads a byte from external memory, the EMIF reads the full width of the memory, and the requested byte is isolated in the DSP.

Figure A–11. Accessing 8-Bit Data in 32-Bit-Wide and 16-Bit-Wide External Memory



A.4.1 8-Bit Data Access of 32-Bit-Wide Memory

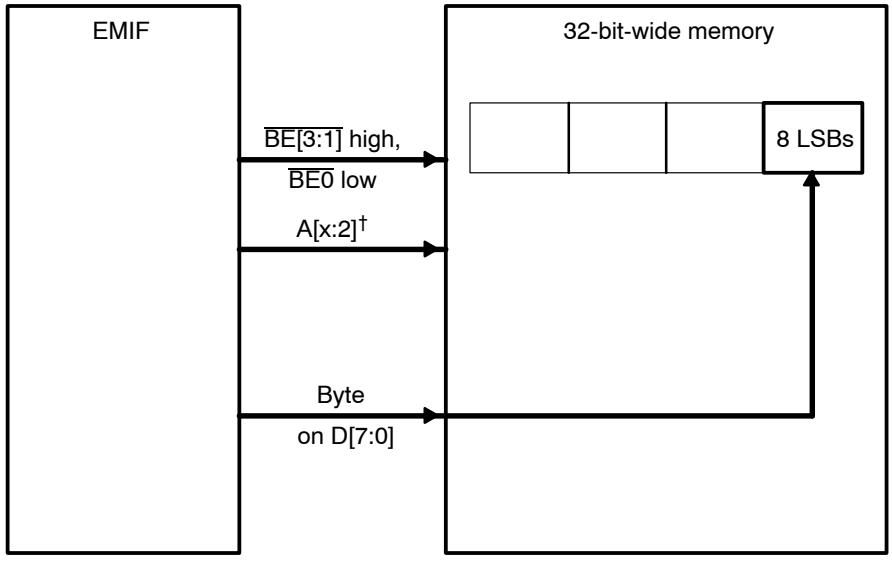
Reading 8-bit data. When reading 8 bits of data from 32-bit-wide memory, the EMIF performs a 32-bit data access (see section A.2.1 on page A-5). The EMIF reads the full width of the memory, and the requested byte is isolated in the DSP.

Writing 8-bit data. When the EMIF writes 8-bit data to 32-bit-wide external memory, the least significant address line required by the external memory chip is A2. However, the EMIF uses bits 1 and 0 of the internal address to determine which byte is loaded, which data lines carry the data, and which byte enable signal is active (see Table A–4). As one example, Figure A–12 shows the EMIF modifying the 8 LSBs (bits 7–0) of a memory location.

Table A–4. The Role of Internal Address Bits 1–0 During an 8-Bit Data Write to 32-Bit-Wide External Memory

Internal Address Bits 1–0	Bits Loaded At Memory Location	Data Lines Used	Byte Enable Signal Levels
00	31–24 (the 8 MSBs)	D[31:24]	$\overline{BE3}$ low (active), others high
01	23–16	D[23:16]	$\overline{BE2}$ low (active), others high
10	15–8	D[15:8]	$\overline{BE1}$ low (active), others high
11	7–0 (the 8 LSBs)	D[7:0]	$\overline{BE0}$ low (active), others high

Figure A–12. Writing to the 8 LSBs of a 32-Bit-Wide External Memory Location



† The least significant address line required is A2. The type of memory accessed determines which other address lines are required.

A.4.2 8-Bit Data Access of 16-Bit-Wide Memory

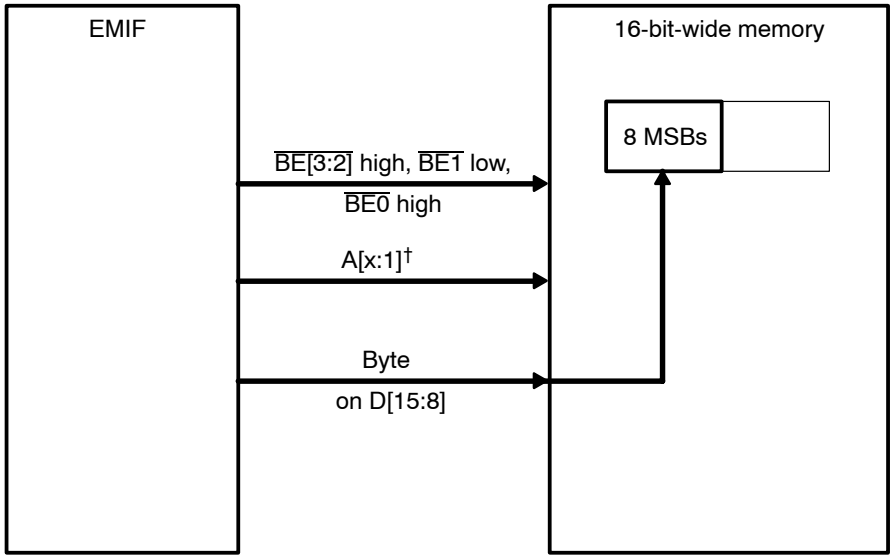
Reading 8-bit data. When the EMIF reads 8-bit data from 16-bit-wide memory, it performs a 16-bit data access (see section A.3.2 on page A-13). The EMIF reads the full width of the memory, and the requested byte is isolated in the DSP.

Writing 8-bit data. When the EMIF writes a byte to 16-bit-wide external memory, the least significant address line required by the external memory chip is A1. However, the EMIF uses bit 0 of the internal address to determine which byte is loaded, which data lines carry the data, and which byte enable signal is active (see Table A–5). Figure A–13 shows the EMIF modifying the 8 MSBs (bits 15–8) of a memory location.

Table A–5. The Role of Internal Address Bit 0 During an 8-Bit Data Write to 16-Bit-Wide External Memory

Internal Address Bit 0	Bits Loaded At Memory Location	Data Lines Used	Byte Enable Signal Levels
0	15–8 (the 8 MSBs)	D[15:8]	$\overline{BE1}$ low (active), others high
1	7–0 (the 8 LSBs)	D[7:0]	$\overline{BE0}$ low (active), others high

Figure A–13. Writing to the 8 MSBs of a 16-Bit-Wide External Memory Location

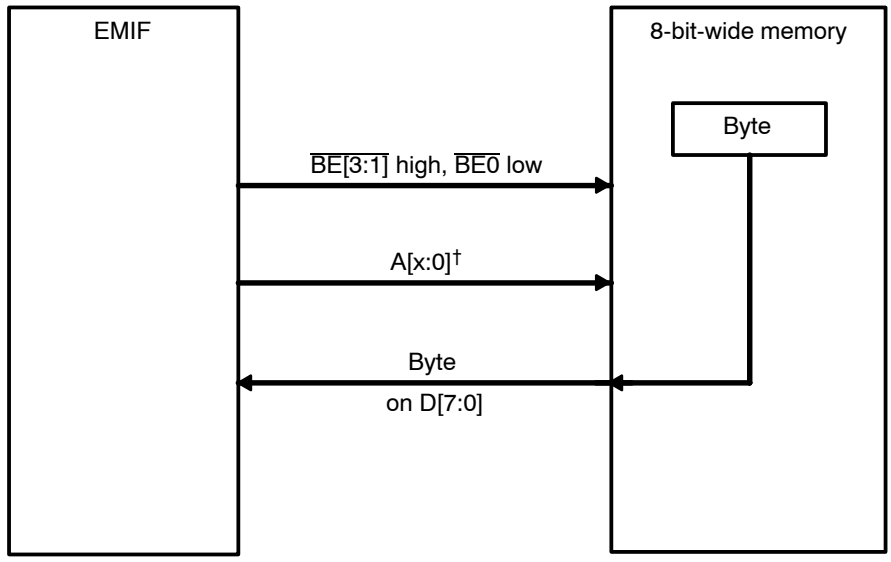


† The least significant address line required is A1. The type of memory accessed determines which other address lines are required.

A.4.3 8-Bit Data Read From 8-Bit-Wide Memory

Figure A–14 illustrates the data transfer involved when the EMIF reads 8-bit data from external memory that is 8 bits wide. The least significant address line required by 8-bit-wide memory is A0. Data bus lines D[7:0] are used to transport the data between the DSP and the external memory. During the access, the EMIF drives $\overline{BE3}$, $\overline{BE2}$, and $\overline{BE1}$ high and drives $\overline{BE0}$ low.

Figure A–14. 8-Bit Data Read From 8-Bit-Wide External Memory



† The least significant address line required is A0. The type of memory accessed determines which other address lines are required.

Appendix B

Revision History

This document was reviewed and no changes were necessary for this version.

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

W