TMS320VC5501/5502 DSP External Memory Interface (EMIF) Reference Guide

Literature Number: SPRU621F
August 2005

Texas Instruments

Preface

Read This First

About This Manual

This manual explains the common operation of the external memory interface (EMIF) in the TMS320VC5501/5502 digital signal processor (DSP).

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.

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.

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 Reference Guide (literature number SPRU317) describes 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

TMS320, TMS320C5000, TMS320C55x, and C55x are trademarks of Texas Instruments.

Trademarks are the property of their respective owners.

Contents

Figures

Tables

Chapter 1

Overview

This chapter provides an overview and describes the common operation of the external memory interface (EMIF).

Topic **Page**

1.1 Overview

The external memory interface (EMIF) supports 8-bit, 16-bit and 32-bit CPU/DMA access interface to a variety of external devices, including:

- Asynchronous devices, including asynchronous SRAM, ROM, Flash with timing parameter programmability
- Synchronous DRAM (SDRAM) running at either the internal or external clock with timing parameter programmability
- Synchronous burst SRAM (SBSRAM) running at either 1x or 1/2x or 1/4x of the internal or external clock rate

The external memory interface services requests of the external bus from seven possible requesting sources:

- Four CPU data busses
- Program fetches from CPU
- Data accesses from the on-chip DMA controller
- SDRAM refresh requests from an internal timer

If multiple requests arrive simultaneously, the EMIF prioritizes them and performs the necessary cycles.

Users can select internal clock or external clock mode for the EMIF clock by tying the EMIFCLKS pin high or low. If the internal clock is selected, it can be set x1, x1/2, or x1/4 CPU clock.

1.2 Resetting the EMIF

A hardware reset using the RESET pin on the device forces all register values to their reset state. During reset, all outputs are driven to their inactive levels, with the exception of the clock outputs (ECLKOUT1 and ECLKOUT2). During active RESET, the clock outputs behave as follows:

- CLKOUT: continues clocking
- ECLKOUTn: the EK1HZ and EK2HZ bits in the EMIF global control register determine the state of ECLKOUT1 and ECLKOUT2 during reset

1.3 HOLD Interface

The EMIF responds to hold requests for the external bus. The hold handshake allows an external device and the EMIF to share the external bus. The handshake mechanism uses two signals:

- $\overline{\text{HOLD}}$: Hold request input. $\overline{\text{HOLD}}$ is synchronized internally to the CPU clock. This synchronization allows an asynchronous input while avoiding metastability. The external device drives this pin low to request bus access. $\overline{\text{HOLD}}$ is the highest priority request that the EMIF can receive during active operation. When the hold is requested, the EMIF stops driving the bus at the earliest possible moment, which may entail completion of the current accesses, device deactivation, and SDRAM bank deactivation. The external device must continue to drive $\overline{\text{HOLD}}$ low for as long as it wants to drive the bus. If any memory spaces are configured for SDRAM, these memory spaces are deactivated and refreshed after $\overline{\text{HOLD}}$ is released by the external master.
- $\overline{\text{HOLDA}}$: Hold acknowledge output. The EMIF asserts this signal active after it has placed its signal outputs in the high-impedance state. The external device can then drive the bus as required. The EMIF places all outputs in the high-impedance state with the exception of $\overline{\text{HOLDA}}$, and the clock outputs (CLKOUT, ECLKOUTx). The EKxHZ bits in the EMIF global control register (GBLCTL2) determine the state of the ECLKOUTx signals while $\overline{\text{HOLDA}}$ is asserted. If any memory spaces are configured for SDRAM, these memory spaces are deactivated and refreshed before $\overline{\text{HOLDA}}$ is asserted to the external master.

During host requests, the refresh counters within the EMIF continue to log refresh requests; however, no refresh cycles can be performed until bus control is again granted to the EMIF when the $\overline{\text{HOLD}}$ input returns to the inactive level. You can prevent an external hold by setting the NOHOLD bit in the EMIF global control register.

1.3.1 Reset Consideration With the Hold Interface

If a hold request is pending upon exiting reset, none of the EMIF output signals are driven active. All output signals stay in a high-impedance state. The $\overline{\text{HOLDA}}$ signal is asserted immediately.

1.4 Boundary Conditions When Writing to EMIF Registers

The EMIF has internal registers that change memory type, asynchronous memory timing, SDRAM refresh, SDRAM initialization (MRS COMMAND), clock speed, arbitration type, HOLD/NOHOLD condition, etc.

The following actions can cause improper data reads or writes:

- Writing to the CE0, CE1, CE2, or CE3 space control registers while an external access to that CE space is active
- Changing the memory type (MTYPE) in the CE space control register while any external operation is in progress (SDRAM type while SDRAM initialization is active)
- Changing the state of NOHOLD in the configuration while HOLD is active at the pin
- Initiating an SDRAM INIT (MRS) while the $\overline{\text{HOLD}}$ input or the $\overline{\text{HOLDA}}$ output is active

The EMIF global control register can be read before the SDRAM INIT bit is set, to determine if the HOLD function is active. It must be read immediately after the SDRAM INIT bit is written to make sure that the two events did not occur simultaneously.

The EMIF global control register has status on the $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$, DMA active access and false access detection.

1.5 Clock Output Enabling

To reduce electromagnetic interference (EMI) radiation, the EMIF allows the disabling (holding high) of CLKOUT, ECLKOUTx. This disabling is performed by setting the CLK2EN, CLK1EN bits to 0 in the EMIF global control register2 (GBLCTL2). The ECLKOUT2 can be configured to run at 1x, 1/2x, or 1/4x, which is the EMIF CLOCK rate for the generic synchronous interface. In addition, the EK1HZ and EK2HZ bits in the GBLCTL configure the output EMIF clock behavior during hold. Table 1−1 summarizes the function of the EKxEN and EKxHZ bits.

Table 1−1. EMIF Output Clock (ECLKOUTx) Operation

EKxEN	EKxHZ	ECLKOUTx Behavior
0	0	ECLKOUTx remains low
0	1	ECLKOUTx low,except during HOLD. In high-impedance during HOLD
1	0	ECLKOUTx clocking
1	1	ECLKOUTx clocking, except during HOLD. In high-impedance during HOLD

1.6 Emulation Halt Operation

The EMIF continues operating during emulation halts. Emulator accesses through the EMIF can work differently than the way the actual device works during EMIF accesses. This discrepancy can cause startup penalties after a halt operation.

1.7 Power Down

Refresh is enabled if EMIF CLOCK is provided, unless the EMIF is in power-down mode. When the EMIF is in power-down mode, the EMIF acts as if it were in reset with ECLKOUTx held low.

Chapter 2

EMIF Operation and Registers

This chapter describes the common operation and registers of the EMIF.

Topic **Page**

2.1 EMIF Signal Descriptions

Table 2−1 offers a complete description of the EMIF signals, while Figure 2−1 shows a block diagram of the EMIF, the interface between external memory and the other internal units of the C5501/5502 DSP.

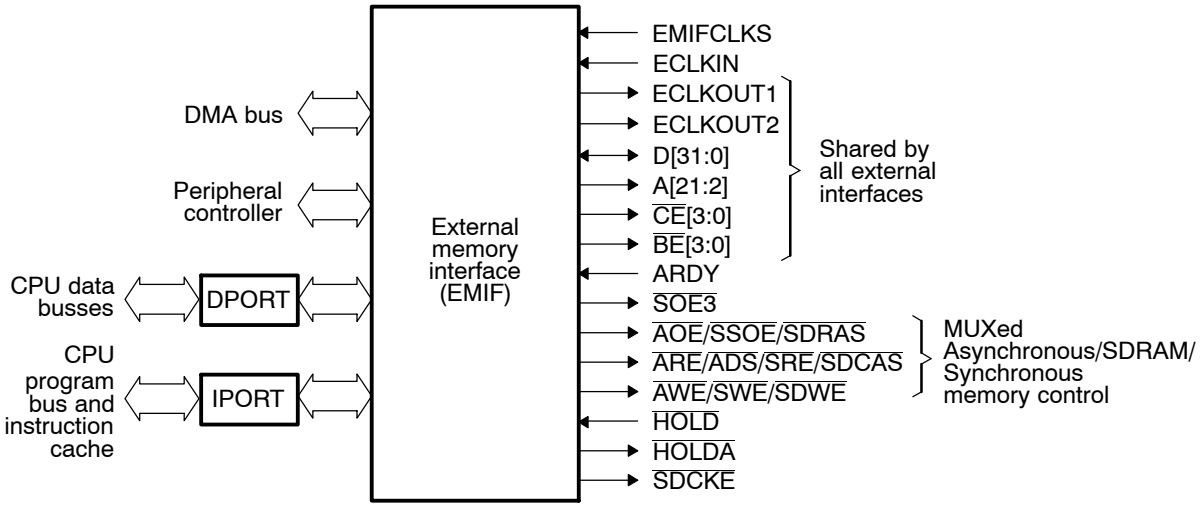
Table 2−1. EMIF Signal Description

EMIF Signal	I/O/Z	Description
D[31:0]	I/OZ	32-bit data bus.
A[21:2]	O/Z	External address bus. Drives the internal address bus, as shown on Table 2−2.
$\overline{\text{CE0}}$	O/Z	Active-low chip select for memory space CE0.
$\overline{\text{CE1}}$	O/Z	Active-low chip select for memory space CE1.
$\overline{\text{CE2}}$	O/Z	Active-low chip select for memory space CE2.
$\overline{\text{CE3}}$	O/Z	Active-low chip select for memory space CE3.
$\overline{\text{BE}}$[3:0]	O/Z	Byte enables. Active-low byte enable. Individual bytes and half-words can be selected for both read and write cycles.
ARDY	I	Ready. Active-high ready input used to insert wait states for slow memories.
$\overline{\text{AOE}}$/$\overline{\text{SSOE}}$/$\overline{\text{SDRAS}}$	O/Z	Active-low output enable for asynchronous/synchronous memory interface or for row address strobe ($\overline{\text{RAS}}$) for SDRAM memory interface.
$\overline{\text{AWE}}$/$\overline{\text{SWE}}$/$\overline{\text{SDWE}}$	O/Z	Active-low write strobe for asynchronous/synchronous/SDRAM memory interface.
$\overline{\text{ARE}}$/$\overline{\text{ADS}}$/$\overline{\text{SRE}}$/$\overline{\text{SDCAS}}$	O/Z	Active-low read strobe for asynchronous memory interface/ address strobe for synchronous memory interface/ column address strobe ($\overline{\text{CAS}}$) for SDRAM memory interface/read enable.
$\overline{\text{SDCKE}}$	O/Z	Active-low output enable for SDRAM.
$\overline{\text{SOE3}}$	O/Z	Active-low output enable for synchronous memory interface for $\overline{\text{CE3}}$ (intended for glueless FIFO interface).
$\overline{\text{HOLD}}$	I	Active-low external bus hold (3-state) request.
$\overline{\text{HOLDA}}$	O	Active-low external bus hold acknowledge.
ECLKIN	I	External EMIF input clock. When selected, this pin provides the clock to the EMIF. Refer to the device-specific data manual for more information on the maximum operating frequency of this input clock.

Table 2–1. EMIF Signal Description (Continued)

EMIF Signal	I/O/Z	Description
ECLKOUT1	O/Z	EMIF output clock at EMIF clock frequency.
ECLKOUT2	O/Z	EMIF output clock at EMIF clock frequency div 1, 2, and 4.
EMIFCLKS	I	EMIF input clock source select. When pulled low, the EMIF operates using an internal clock. When pulled high, the EMIF clock is taken from an external source through the ECLKIN pin.

Figure 2–1. EMIF Block Diagram



2.2 External Address Outputs

The VC5501/5502 EMIF works with two buses: an internal address bus A[19:0] and external address bus A[21:2]. The address shown on the external address bus is a shifted version of the internal address bus. The EMIF automatically shifts its internal address bus to match the data bus size of the memory device being used.

The mapping of internal address lines to external address pins after shifting is shown in Table 2–2 for the asynchronous and SBSRAM configurations. The address shifting for the SDRAM configuration is described in Section 2.6.4.

Table 2–2. Internal Address Mapping and EMIF Address Pins for ASYNC and SBSRAM Configuration

Memory Type	External Data Bus Width	Internal Address Routed to Pins A[21:2]
ASYNC Memory	8-bit	Internal Address A [19:0]
	16-bit	Internal Address A [20:1]
	32-bit	Internal Address A [21:2]
SBSRAM	8-bit	Internal Address A [19:0]
	16-bit	Internal Address A [20:1]
	32-bit	Internal Address A [21:2]

The address bus of the external memory device should always be connected to the EMIF address bus starting with pin A2. For example, address lines A[n:0] of an 8-, 16- or 32-bit asynchronous memory device, where n is the number of address lines on the memory device, should be connected to address lines A[n+2:2] on the EMIF.

2.3 External CE Outputs

Table 2–3 shows how the EMIF can be configured up to four memory spaces (CE).

Table 2–3. CE Output Configuration

CE Configuration (Number of CEs)	Available CEs	CE Selected by Address Bits	Size Per CE Space
4	CE0/1/2/3	[23:22]	4M Byte
2	CE0/2	[23]	8M Byte
1	CE0	All 24	16M Byte

2.4 Asynchronous Interface

The asynchronous interface offers configurable memory cycle types to interface with a variety of asynchronous memory types including SRAM, EPROM, and Flash memory, as well as FPGA and ASIC designs.

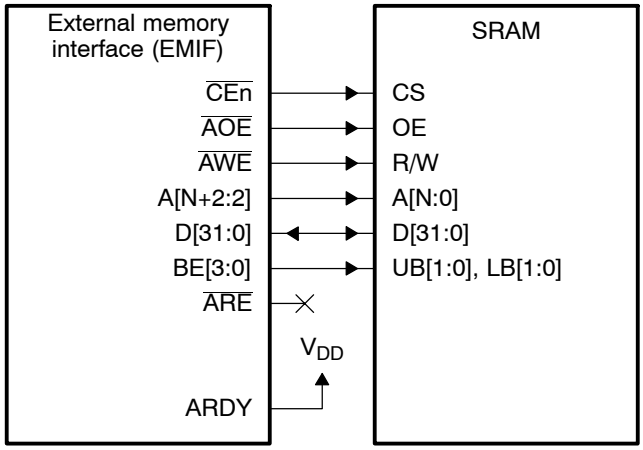
Table 2−4. EMIF Asynchronous Interface Pins

EMIF Signal	Function
$\overline{\text{AOE}}$	Output enable. Active (low) during the entire period of a read access.
$\overline{\text{AWE}}$	Write enable. Active (low) during a write transfer strobe period.
$\overline{\text{ARE}}$	Read enable. Active (low) during a read transfer strobe period.
ARDY	Ready. Input used to insert wait states into the memory cycle.

Figure 2−2 shows an interface to standard SRAM. The C5501/5502 allows a width of less than 32 bits on any CE space, as shown in the MTYPE description of the CExCTL register. The asynchronous interface signals on the C5501/5502 are combined with the SDRAM and SBSRAM memory interface. It has also been enhanced to allow for longer read hold time, and the 8- and 16-bit interface modes have been extended to include writable asynchronous memories, instead of ROM devices. A programmable turnaround time (TA) also allows the user to control the number of cycles between a read and a write to avoid bus contention. The asynchronous interface has these features:

- Interface width - 32-bit, 16-bit, 8-bit
- Internal synchronization - ECLKOUT1
- Control signals - ASRAM control signals are multiplexed with SDRAM and programmable synchronous control signals
- Memory endianess - Supports only big endian

Figure 2–2. EMIF-to-SRAM Interface



2.4.1 Address Shift

The EMIF automatically shifts its internal address bus to match the data bus size of the memory device being used. The address shifting is described in Table 2–5 for the asynchronous configuration. The address bus of the external memory device should always be connected to the EMIF address bus starting with pin A2. For example, address lines A[n:0] of an 8-, 16- or 32-bit asynchronous memory device, where n is the number of address lines on the memory device, should be connected to address lines A[n+2:2] on the EMIF.

Table 2–5. Internal Address Mapping to EMIF Address Pins for ASYNC Configuration

Memory Type	External Data Bus Width	Internal Address Routed to Pins A[21:2]
ASYNC Memory	8-bit	Internal Address A [19:0]
	16-bit	Internal Address A [20:1]
	32-bit	Internal Address A [21:2]

2.4.2 Programmable Asynchronous Memory Parameters

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

- Setup: The time between the beginning of a memory cycle ($\overline{CE}$ low, address valid) and the activation of the read or write strobe
- Strobe: The time between the activation and deactivation of the read ($\overline{ARE}$) or write strobe ($\overline{AWE}$)
- Hold: The time between the deactivation of the read or write strobe and the end of the cycle (which is an address change)

These parameters are programmable in terms of ECLKOUT1 clock cycles via fields in the EMIF CE space control registers. Separate setup, strobe, and hold parameters are available for read and write accesses. The SETUP and STROBE fields have a minimum count of 1, and because of this, a value of 0 in these fields is still interpreted as a 1 by the C5501/5502. For the first access in a set of consecutive accesses for a single access, the setup period has a minimum of 2. HOLD can be set to 0 cycles. Table 2−6 summarizes the setup behavior.

Table 2−6. Setup Time in ECLKOUT1 Cycles

Setup Configuration	Actual Setup for First Access	Actual Setup After First Access
0	2	1
1	2	1
2	2	2
3	3	3
4	4	4

Figure 2−3 illustrates an asynchronous single read cycle from 32-bit-wide memory with a setup/strobe/hold timing of 2/3/1. In this case, the asynchronous read proceeds as follows:

- At the beginning of the setup period:
  - $\overline{CE}$ becomes active low
  - $\overline{AOE}$ becomes active low
  - $\overline{BE}$[3:0] becomes valid
  - Address bus becomes valid
  - For first access, setup has a minimum value of 2

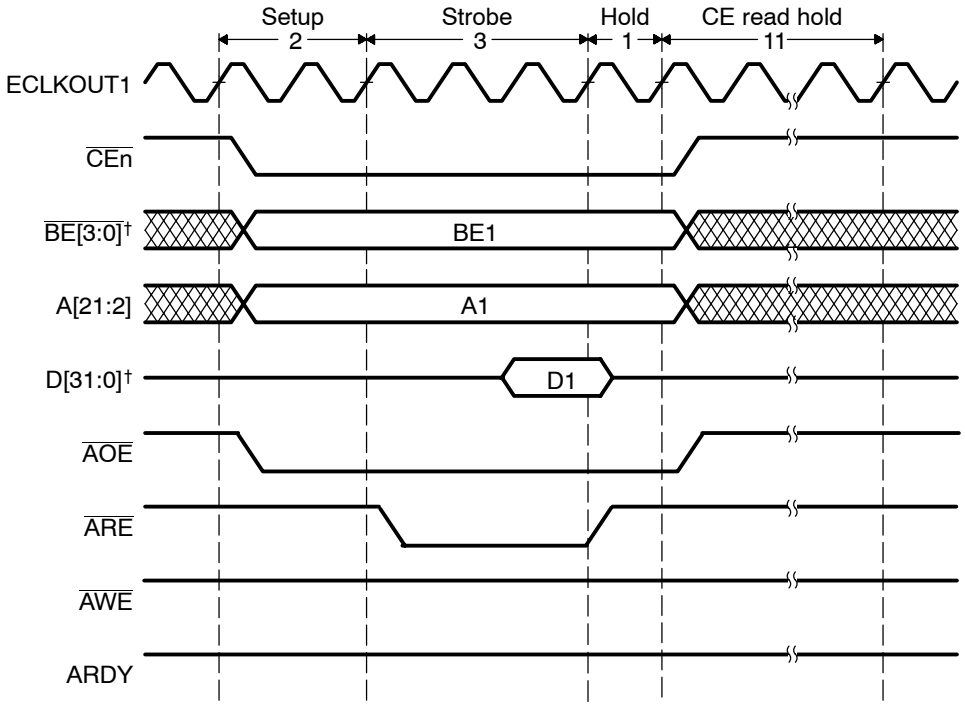
- At the beginning of a strobe period:
  - $\overline{ARE}$ becomes active low
- At the beginning of a hold period:
  - $\overline{ARE}$ is pulled inactive high
  - Data lines are sampled on the DSP clock rising edge concurrent with the beginning of the hold period (end of the strobe period), just prior to the $\overline{ARE}$ low-to-high transition
- At the end of the hold period:
  - $\overline{AOE}$ becomes inactive at the end of a burst access or at the end of a single access.
  - An automatic CE read hold period is added after the end of the programmed read hold period. A new read access begins with the programmed read setup period after the end of the CE read hold period

Note:

The EMIF always reads 32-bits from external memory, regardless of the size for the original data request. Refer to section 2.9, Data Accesses, for more information.

The CE read hold period is added automatically by the EMIF; it is not programmable. Depending on the size of the memory, a read from external memory may be composed of several read cycles. For example, a read from 16-bit-wide external memory is composed of two 16-bit read cycles. The CE read hold period and programmed read hold period for the last read cycle always add up to 12 EMIF clock cycles. For example, if the programmed read hold period is set to 5, the CE hold period will be 7. The programmed setup and strobe periods do not affect the CE read hold period. Figure 2–4 and Figure 2–5 show examples of single reads from 16- and 8-bit-wide asynchronous memory.

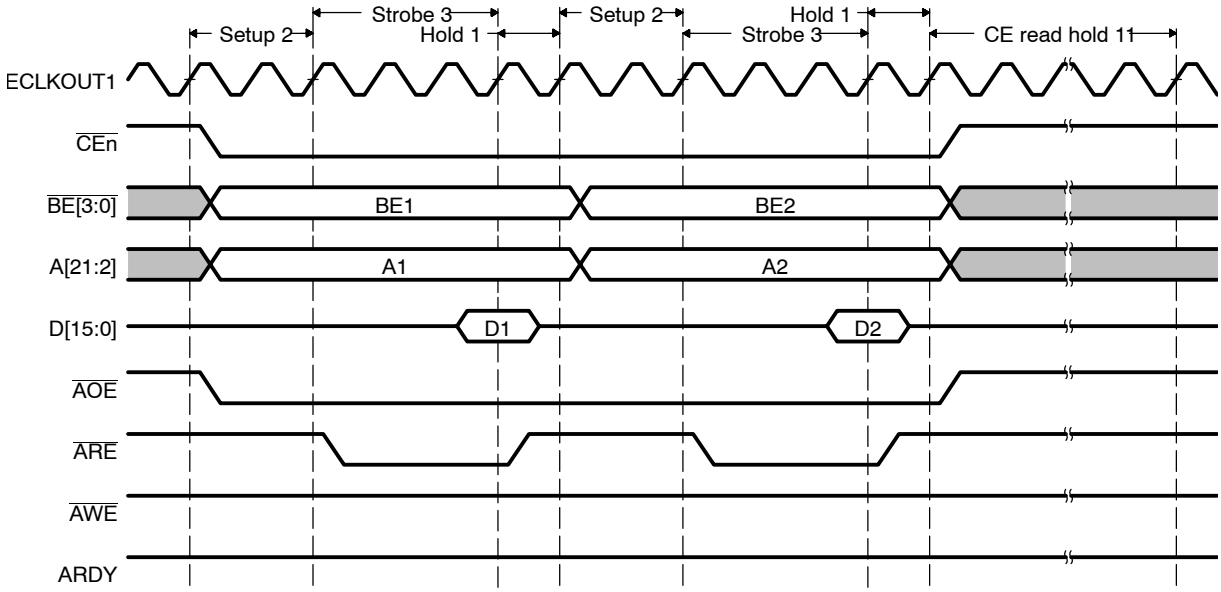
Figure 2–3. Asynchronous Read Timing Example (32-Bit-Wide Memory)



† The EMIF always reads 32-bits from external memory, therefore, all byte enable signals and data lines will be active during a read cycle.

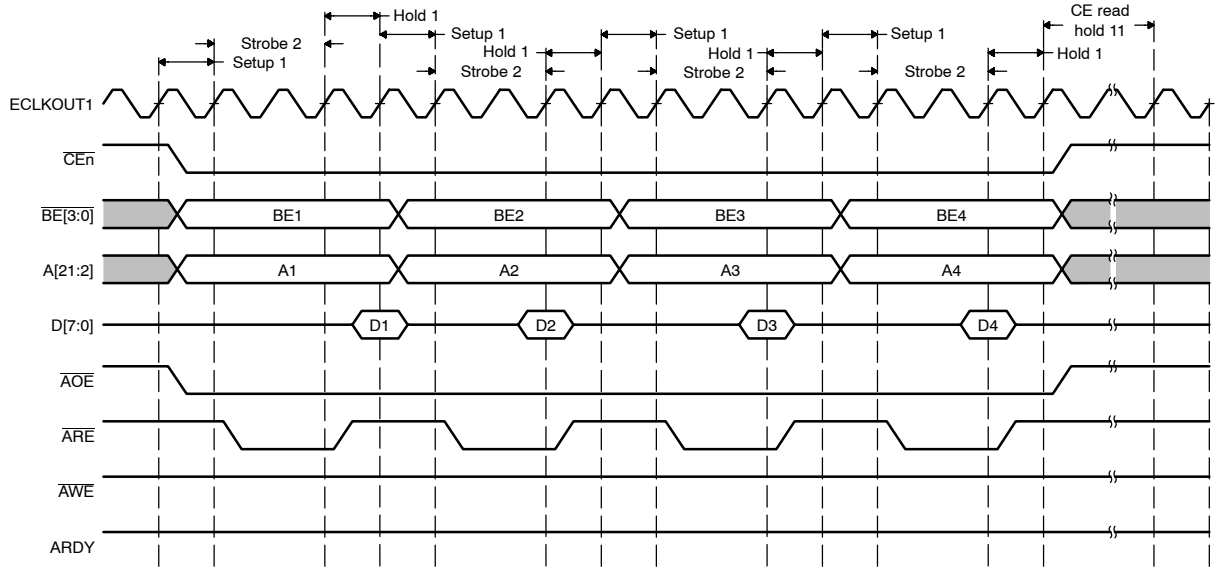
When doing a single read from 16-bit-wide memory, the EMIF reads 32-bits from external memory by executing two 16-bit read cycles. Only the programmed hold period of the second 16-bit access affects the CE read hold period; these two always add up to 12. An example of a single read from 16-bit-wide memory is illustrated in Figure 2–4.

Figure 2–4. Asynchronous Read Timing Example (16-Bit-Wide Memory)



When doing a single read from 8-bit-wide memory, the EMIF reads 32-bits by executing four 8-bit read cycles. The programmed read hold period of the last 8-bit access and the CE read hold period always add up to 12. Figure 2–5 shows an example of a single read from 8-bit-wide memory.

Figure 2–5. Asynchronous Read Timing Example (8-Bit-Wide Memory)



2.4.3 Asynchronous Writes

Figure 2−6 illustrates an asynchronous single write cycle from 32-bit-wide memory with a setup/strobe/hold of 2/3/1. In this case, the asynchronous write proceeds as follows.

- At the beginning of the setup period:
  - $\overline{CE}$ becomes active low
  - $\overline{BE}$[3:0] becomes valid
  - Address bus becomes valid
  - Data bus becomes valid
  - For the first access, setup has a minimum value of 2; after the first access, setup has a minimum value of 1
- At the beginning of a strobe period:
  - $\overline{AWE}$ becomes active low.
- At the beginning of a hold period:
  - $\overline{AWE}$ is pulled inactive high
  - Data lines are sampled on the ECLKOUT1 rising edge concurrent with the beginning of the hold period (end of the strobe period), just prior to the $\overline{AWE}$ low-to-high transition
- At the end of the hold period:
  - Data go into the high-impedance state only if another write to the same $\overline{CE}$ is not scheduled for the next cycle
  - An automatic CE write hold period is added after the end of the programmed write hold period. A new access begins with the programmed write setup period after the end of the CE write period, 2 in this case

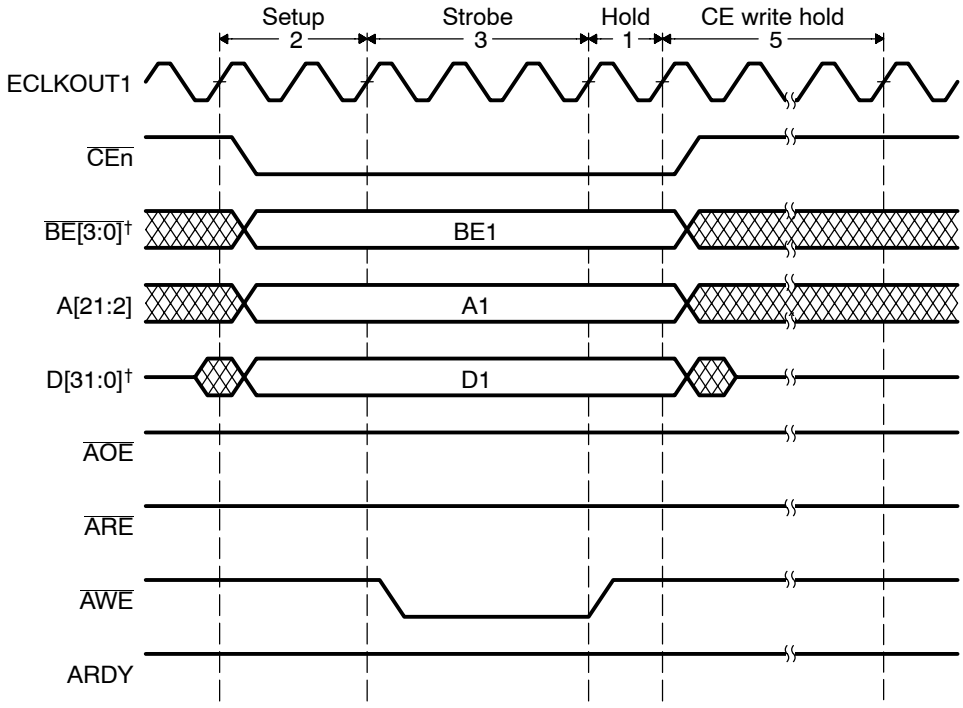
The CE write hold period is added automatically by the EMIF for every write access; it is not programmable. Depending on the size of the memory and the size of the write request, a write to external memory may be composed of several write cycles. For example, a 16-bit write to 8-bit-wide external memory is composed of two 8-bit write cycles. Together, all programmed write setup, strobe, and hold periods, and the CE write hold period add up to a minimum of 11 EMIF clock cycles (9 if write posting is enabled). Any size request can be serviced in a single external memory access when using 32-bit-wide memory. Therefore, if the programmed setup, strobe, and hold periods were 1, 5, and 2, respectively, the CE write hold period would be 3 and the write access request would take a total of 11 EMIF clock cycles (see Figure 2−6).

On the other hand, if the programmed setup, strobe, and hold periods were 10, 10, and 7, the CE write hold period would be 0 and the write access request would take 27 EMIF clock cycles.

As another example, consider writing 32-bits to 16-bit-wide external memory. This operation requires two 16-bit writes; a total of two write setup periods, two strobe periods, two hold periods, and one CE write hold period. As mentioned before, all programmed write setup, strobe, and hold periods and the CE write hold period add up to a minimum of 11 EMIF clock cycles (9 if write posting is enabled). An example of a single 32-bit write to 16-bit external memory is shown in Figure 2−7.

Figure 2−8 illustrates a single 16-bit write to 8-bit-wide memory. Note that even with a minimum setup, strobe, and hold of 1, a 32-bit write to 8-bit memory would take 12 cycles, therefore the CE write hold period will always be zero.

Figure 2−6. Asynchronous Write Timing Example (32-Bit-Wide Memory)



† The number of byte enable signals and data lines that are active depends on the size of data being written to memory.

Figure 2–7. 32-Bit Asynchronous Write Timing Example (16-Bit-Wide Memory)

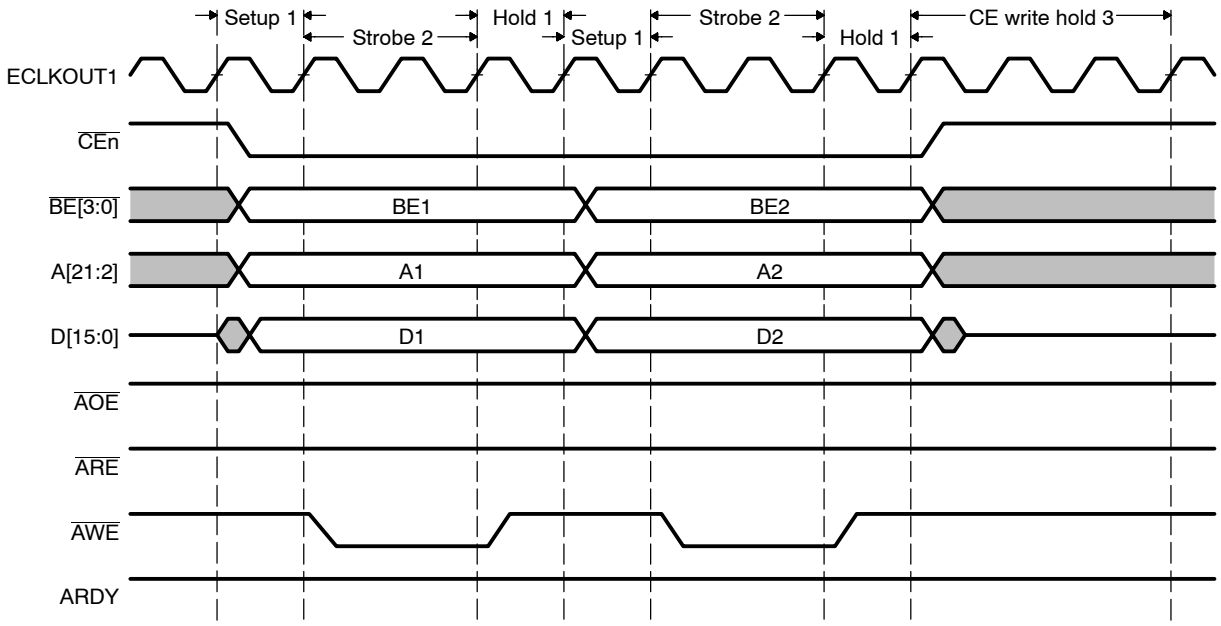
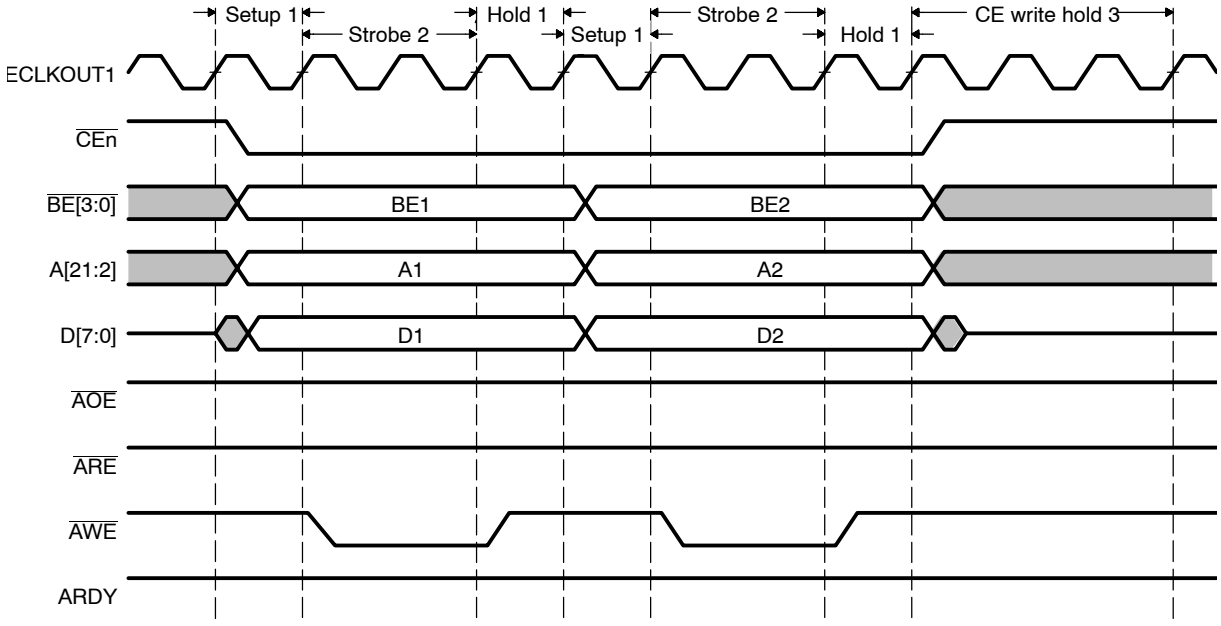


Figure 2–8. 16-Bit Asynchronous Write Timing Example (8-Bit-Wide Memory)

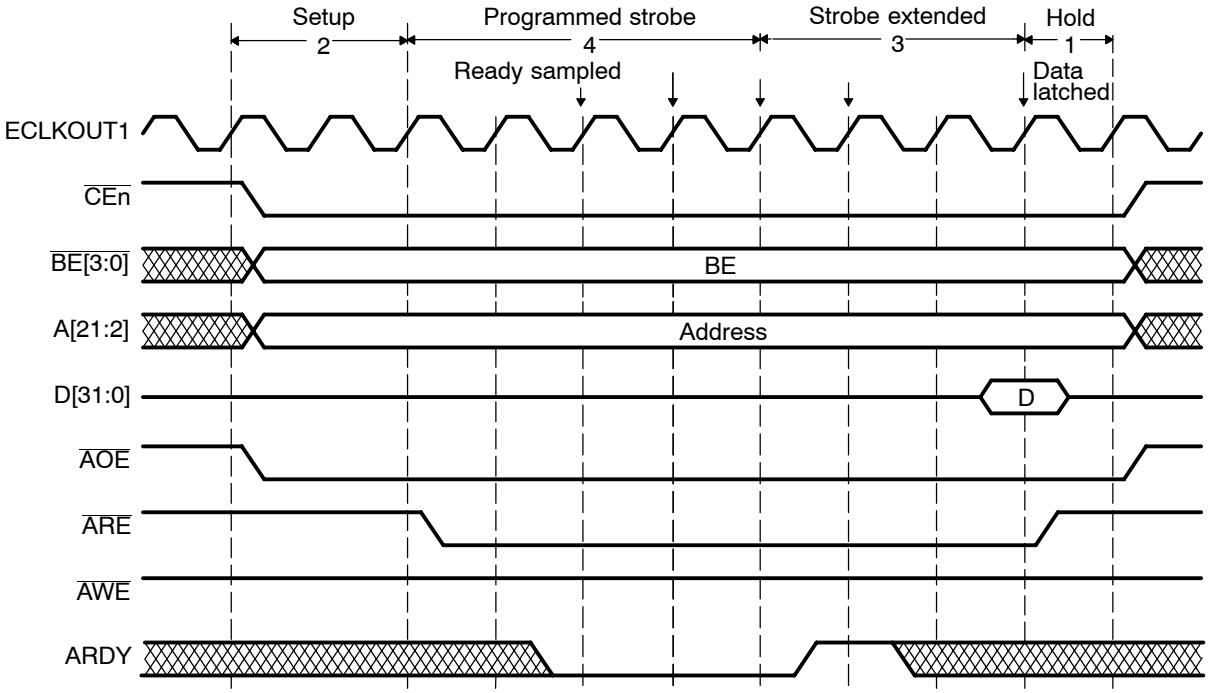


2.4.4 Ready Input

In addition to programmable access shaping, you can insert extra cycles into the strobe period by deactivating the ARDY input. The ready input is internally synchronized to ECLKOUT1. This synchronization allows an asynchronous ARDY input while avoiding metastability.

If ARDY is low on the second rising edge of ECLKOUT before the end of the programmed strobe period (Figure 2−9), then the strobe period is extended by one ECLKOUT cycle. For each subsequent ECLKOUT rising edge that ARDY is sampled low, the strobe period is extended by one ECLKOUT cycle. Thus, to effectively use CE to generate ARDY inactive with external logic the minimum of SETUP and STROBE should be 3.

Figure 2−9. Asynchronous Read Timing With Ready Input Example



2.5 SBSRAM Interface

The EMIF interfaces directly to 32-bit wide industry standard synchronous burst SRAMs. SBSRAMs are available in both flow through and pipeline, however, the C5501/5502 interfaces only to pipeline SBSRAM, which has the capability to operate at higher frequencies with sustained throughput. Figure 2−10 illustrates the EMIF to SBSRAM interface.

The SBSRAM interface can run at ECLKOUT1 or ECLKOUT2 clock speed.

The three SBSRAM control signals are latched by the SBSRAM on the rising EMIF clock edge to determine the current operation. These signals are only valid if the chip select line for the SBSRAM is low. The $\overline{\text{ADV}}$ signal of the SBSRAM is used to allow the SBSRAM device to generate addresses internally for interfacing to controllers which cannot provide addresses quickly enough. However, the C5501/5502 EMIF does not need to use this signal because it generates the addresses at the required rate.

Table 2−7. TMS320VC5501/5502 Programmable Synchronous Pins

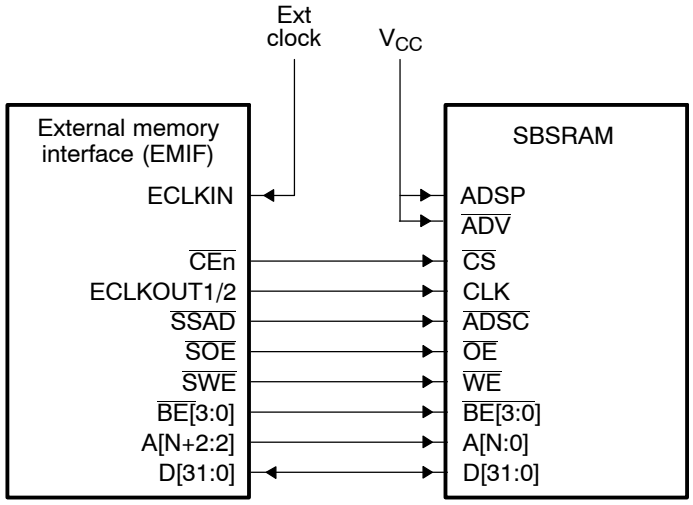
Pin	Description
$\overline{\text{SADS}}$/$\overline{\text{SRE}}$	Address strobe/read enable (selected by RENEN)
$\overline{\text{SOE}}$	Output enable
$\overline{\text{SOE3}}$	Output enable for CE3. The SOE3 pin is not multiplexed with other signals. (useful for glueless FIFO interface)
$\overline{\text{SWE}}$	Write enable
ECLKOUT1	Synchronous interface clock, runs at 1x EMIF input clock rate
ECLKOUT2	Synchronous interface clock, runs at 1x, 1/2x, or 1/4x EMIF input clock rate

The C5501/5502 interface does not support the burst feature of the SBSRAM. On the C5501/5502, an address is strobed into the SBSRAM on every cycle. After performing a read or write command, the C5501/5502 EMIF issues a deselect command to the SBSRAM if no accesses are pending to that CE space. The C5501/5502 also supports programmable read and write latency to allow a flexible interface to different types of synchronous memories.

To summarize the features of the SBSRAM interface:

- Interface width - 32-bit, 16-bit, 8-bit
- SBSRAM clock - ECLKOUT1 or ECLKOUT2
- SBSRAM control signals - multiplexed with SDRAM and Async control signals
- Burst mode - not supported. Performs bursts by issuing back-to-back commands. Still issues deselect command.
- Programmable latency - read, write

Figure 2−10. EMIF-to-SBSRAM Interface



2.5.1 Address Shift

The EMIF automatically shifts its internal address bus to match the data bus size of the memory device being used. The address shifting is described in Table 2−8 for the SBSRAM configuration. The address bus of the external memory device should always be connected to the EMIF address bus starting with pin A2. For example, address lines A[n:0] of an 8-, 16- or 32-bit SBSRAM memory device, where n is the number of address lines on the memory device, should be connected to address lines A[n+2:2] on the EMIF.

Table 2−8. Internal Address Mapping to EMIF Address Pins for SBSRAM Configuration

Memory Type	External Data Bus Width	Internal Address Routed to Pins A[21:2]
SBSRAM	8-bit	Internal Address A [19:0]
	16-bit	Internal Address A [20:1]
	32-bit	Internal Address A [21:2]

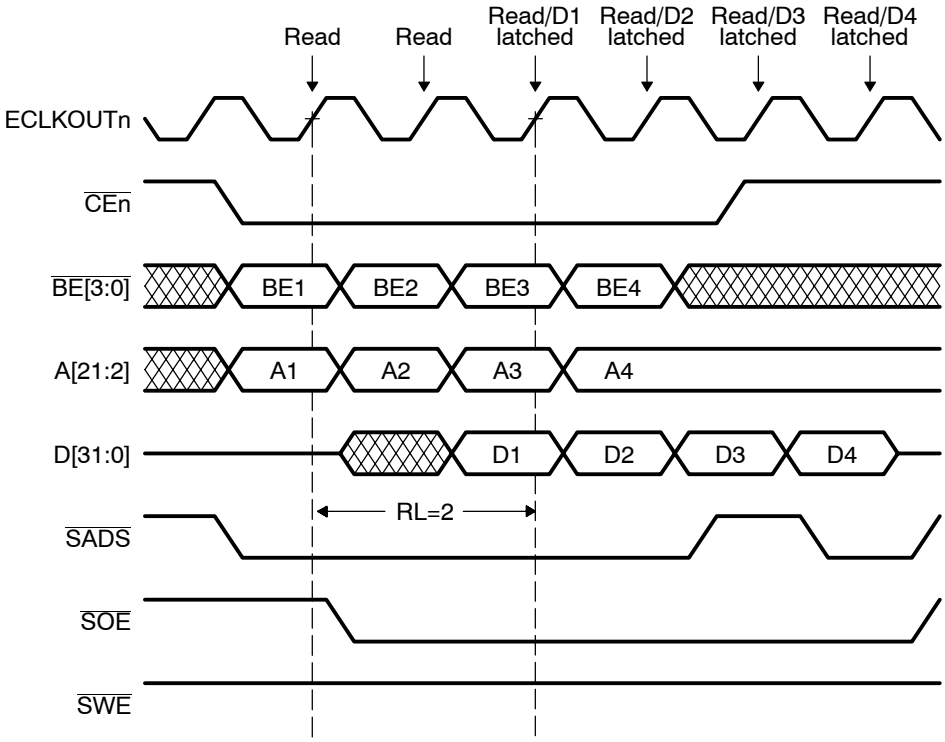
2.5.2 SBSRAM Read

Figure 2−11 shows the timing for C5501/5502 four-word read with a two-cycle read latency. Every access strobes a new address into the SBSRAM, indicated by the $\overline{\text{SADS}}$ strobe low. The C5501/5502 EMIF issues a deselect cycle at the end of the burst transfer.

For the standard SBSRAM interface, the following fields in the CExSEC register must be set to their default state:

- SYNCRL = 10b; 2 cycle read latency
- SYNCWL = 00b; 0 cycle write latency
- CEEXT = 0; CE goes inactive after the final command has been issued
- RENEN = 0; $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ signal acts as $\overline{\text{SADS}}$ signal

Figure 2−11. SBSRAM Read Timing Example

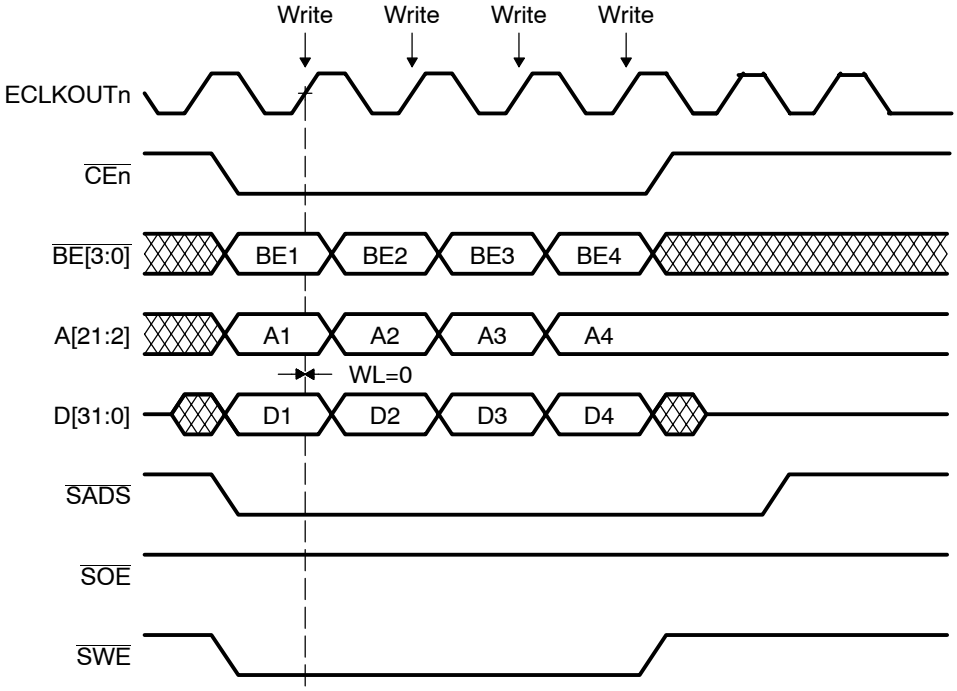


2.5.3 SBSRAM Write

Figure 2−12 shows a C5501/5502 four-word write to SBSRAM. Every access strobes a new address into the SBSRAM. The C5501/5502 EMIF issues a deselect cycle at the end of the burst transfer.

Refer to section 2.5.2 for the CExSEC register setting for the C5501/5502 SBSRAM write interface.

Figure 2−12. SBSRAM Write Timing Example



2.6 SDRAM Interface

The EMIF supports an interface to 16-Mbit, 64-Mbit, 128-Mbit, and 256-Mbit SDRAM, offering an interface to high speed and high-density memory. It supports SDRAM commands shown in Table 2–9.

The EMIF allows programming of the addressing characteristics of the SDRAM, including the number of column address bits (page size), the row address bits (pages per bank) and banks (maximum number of pages which can be opened). Using this information, the EMIF is able to open up to four pages of SDRAM simultaneously. The pages can all be different banks of a single CE space, or distributed across multiple CE spaces. Only one page can be opened per bank at a time. The EMIF can interface to any SDRAM that has 8 to 10 column address pins, 11 to 13 row address pins, and two or four banks.

The EMIF supports the SDRAM self refresh mode, and support the least recently used (LRU) page replacement strategy instead of random replacement strategy for better performance.

Table 2–9. EMIF SDRAM Commands

Command	Function
DCAB	Deactivate (also known as pre-charge) all banks
DEAC	Deactivate a single bank
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	Mode register ret; configures SDRAM mode register
REFR	Auto refresh cycle with internal address
SLFREFR	Self-refresh mode

Table 2−10. True Table of SDRAM Commands

SDRAM	CKE	$\overline{CS}$	$\overline{RAS}$	$\overline{CAS}$	$\overline{W}$	A[19:16]	A[15:11]	A10	A[9:0]
EMIF	**SDCKE**	$\overline{CE}$	$\overline{SDRAS}$	$\overline{SDCAS}$	$\overline{SDWE}$	**A[21:18]**	**A[17:13]**	**A[12]**	**A[11-2]**
ACTV	H	L	L	H	H	0001b or 0000b	Bank/ Row	Row	Row
READ	H	L	H	L	H	X	Bank/ Col	L	Col
WRT	H	L	H	L	L	X	Bank/ Col	L	Col
MRS	H	L	L	L	L	L	L/Mode	Mode	Mode
DCAB	H	L	L	H	L	X	X	H	X
DEAC	H	L	L	H	L	X	Bank/X	L	X
REFR	H	L	L	L	H	X	X	X	X
SLFREFR	L	L	L	L	H	X	X	X	X

Legend: Bank =Bank address; Row = Row address; Col = Column address; L = 0b = Low; H = 1b = High, Mode = Mode select; X = Previous value

Table 2−11. C5501/5502 EMIF Compatible SDRAM

SDRAM Size	Banks	Width	Depth	Max Device/ CE	Addressable Space, Bytes	Memory	Column Address	Row Address	Bank Select	Pre-Charge
16M-bit	2	X4	2M	8	16M	SDRAM	A9-A0	A10-A0	A11	A10
						EMIF	A11-A2	A12-A2	A13	A12
	2	X8	1M	4	8M	SDRAM	A8-A0	A10-A0	A11	A10
						EMIF	A10-A2	A12-A2	A13	A12
	2	X16	512K	2	4M	SDRAM	A7-A0	A10-A0	A11	A10
						EMIF	A9-A2	A12-A2	A13	A12
64M-bit	4	X4	4M	8	64M	SDRAM	A9-A0	A11-A0	A13-A12	A10
						EMIF	A11-A2	A13-A2	A15-A14	A12
	4	X8	2M	4	32M	SDRAM	A8-A0	A11-A0	A13-A12	A10
						EMIF	A10-A2	A13-A2	A15-A14	A12
	4	X16	1M	2	16M	SDRAM	A7-A0	A11-A0	A13-A12	A10
						EMIF	A9-A2	A13-A2	A15-A14	A12
	4	X32	512K	1	8M	SDRAM	A7-A0	A10-A0	A12-A11	A10
						EMIF	A9-A2	A12-A2	A14-A13	A12
128M-bit	4	X8	4M	4	64M	SDRAM	A9-A0	A11-A0	A13-A12	A10
						EMIF	A11-A2	A13-A2	A15-A14	A12
	4	X16	2M	2	32M	SDRAM	A8-A0	A11-A0	A13-A12	A10
						EMIF	A10-A2	A13-A2	A15-A14	A12
	4	X32	1M	1	16M	SDRAM	A7-A0	A11-A0	A13-A12	A10
						EMIF	A9-A2	A13-A2	A15-A14	A12
256M-bit	4	X8	8M	4	128M	SDRAM	A9-A0	A12-A0	A14-A13	A10
						EMIF	EA11-EA2	EA14-EA2	EA16-EA15	EA12
	4	X16	4M	2	64M	SDRAM	A8-A0	A12-A0	A14-A13	A10
						EMIF	EA10-EA2	EA14-EA2	EA16-EA15	EA12
	4	X32	2M	1	32M	SDRAM	A8-A0	A11-A0	A13-A12	A10
						EMIF	EA10-EA2	EA13-EA2	EA15-EA14	EA12

Figure 2−13. EMIF to 64-Mbit SDRAM Interface Using 32-Bit Wide Memory Chip

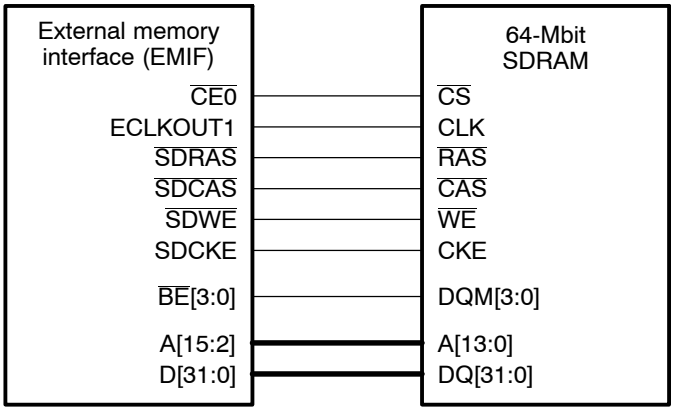


Table 2−12. C5501/5502 SDRAM Pin Summary

EMIF Signal	SDRAM Signal	SDRAM Function
A12	A10	Address line A10/autoprecharge disable.
$\overline{\text{SDRAS}}$	$\overline{\text{RAS}}$	Row address strobe and command input. Latched by the rising edge of CLK to determine current operation. Valid only if $\overline{\text{CS}}$ is active (low) during that clock edge.
$\overline{\text{SDCAS}}$	$\overline{\text{CAS}}$	Column address strobe and command input. Latched by the rising edge of CLK to determine current operation. Valid only if $\overline{\text{CS}}$ is active (low) during that clock edge.
$\overline{\text{SDWE}}$	$\overline{\text{WE}}$	Write strobe and command input. Latched by the rising edge of CLK to determine current operation. Valid only if $\overline{\text{CS}}$ is active (low) during that clock edge.
$\overline{\text{BE}}$x	DQMx	Data/Output mask. DQM is an input/output buffer control signal. When high, disables writes and places outputs in high state during reads. DQM a 2-CLK cycle latency on reads and no latency on writes. DQM pins serve essentially as byte strobes and are connected to $\overline{\text{BE}}$ outputs.
$\overline{\text{CE}}$x	nCS	Chip select and command enable. NCS must be active (low) for a command to be clocked into the SDRAM.
SDCKE	CKE	CKE clock enable**.** SDCKE is connected to CKE to minimize SDRAM power consumption when self-refresh mode is enabled.
ECLKOUT1	CLK	SDRAM clock input. ECLKOUT1 runs at EMIF input clock (ECLKIN, CPU clock)

2.6.1 C5501/5502 SDRAM Interface Summary

- Interface width – 32-bit, 16-bit, 8-bit
- SDRAM clock – ECLKOUT1
- Registers for SDRAM timing parameters – SDCLT, SDTIM, SDEXT
- SDRAM control signals – multiplexed with ASYNC and programmable sync control signals
- Number of open pages – four open pages in any CE space
- Programmable SDRAM configuration – column, row, and bank size
- Burst mode – support SDRAM burst mode with a 4-word burst
- Background refresh – No
- Precharge pin – A12
- SDRAM self-refresh – Yes
- Page replacement – LRU

2.6.2 SDRAM Initialization

The EMIF performs the necessary functions to initialize SDRAM if any of the CE spaces are configured for SDRAM. A SDRAM initialization is requested by a write to the INIT register. This should not be done if an SDRAM access is occurring.

The actual sequence of events of an initialization is as follows:

1) Send a DCAB command to all CE spaces configured as SDRAM
2) Send eight REFR refresh commands
3) Send an MRS command to all CE spaces configured as SDRAM

The DCAB cycle is performed immediately after reset provided that the $\overline{\text{HOLD}}$ input is not active. If $\overline{\text{HOLD}}$ is active, the DCAB command is not performed until the hold condition is removed. Therefore, the external requester should not attempt to access any SDRAM banks if SDRAM initialization has not been done once, unless the external requester will perform SDRAM initialization by itself.

2.6.3 Monitoring 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 the course of an access. To accomplish this monitoring, the EMIF stores the address of the open page, and performs compares against that address for subsequent accesses to the SDRAM bank. This storage and comparison is performed independently for each CE space.

For the C5501/5502 device, ending the current access is not a condition that forces the active SDRAM row to be closed. The EMIF leaves the active row open until it becomes necessary to close it. This decreases the deactivate/reactivate overhead and allows the interface to capitalize fully on the address locality of memory access.

The C5501/5502 SDRAM paging scheme supports up to four pages of SDRAM which can be opened simultaneously in one CE space or spread across multiple CE spaces. For example, two pages can be open in CE0 and CE2, or four pages can be open in CE0. The combination controls which logical address bits are compared to determine if a page is open: SDCSZ (which controls the number of column address bits, or NCB), SDRSZ (which controls the number of row address bits, or NRB), and SDBSZ (which controls the number bank address bits, or NBB). The number of column address bits controls the number of the least significant address stored in the page register.

The page register always stores 16 bits of address (instead of being limited by the number of row address bits plus the number of bank address bits (NRB+NBB)). Therefore, logical address bits above the bank address bits are used as part of the page comparison. Also, address bits above the bank address bits are used when issuing the row/column commands to the external SDRAM. This allows more flexible designs and external visibility into the internal address aliasing. For 32-, 16-, 8-bit interfaces on EMIF, the BE portion of the logical address is reduced to 2 bits for 32-bit SDRAM, 1 bit for 16-bit SDRAM, and 0 bit for 8-bit SDRAM. The NCB/NRB/NBB (and page register) shift accordingly.

The C5501/5502 EMIF employs a least recently used (LRU) page replacement strategy when necessary. This occurs when the total number of external SDRAM banks (not devices) is greater than four, since the EMIF only contains four page registers. This can occur when multiple CE spaces of SDRAM are used. When the number of the total banks of SDRAM is less than or equal to four, then the page replacement strategy is fixed, since SDRAM requires that only 1 page can be open within a given bank. If a page miss is detected either during an access where a different page was previously accessed in the same CE space (fixed replacement), or if a page must be closed within a different CE space to allow a page register to be assigned for the current access (LRU replacement), the C5501/5502 performs a DEAC command and starts a new row access.

2.6.4 Address Shift

The EMIF interface shifts the internal address based on the data bus width of the external memory device being used to select the correct row and column. Table 2−13 shows the translation between bits of the byte address and how they appear on the address pins for row and column addresses on the C5501/5502. SDRAMs use the address inputs for control as well as address.

The following factors apply to the address shifting process for the C5501/5502:

- The address shift is controlled completely by the column size field (SDCSZ), and is unaffected by the bank and row size fields. The bank and row size are used internally to determine whether a page is opened.
- The address bits corresponding to the bank select bits are latched internally by the SDRAM controller during a RAS cycle. The bank select bits are A[13+n:13] for SDRSZ = 00b (11 row pins), A[14+n:14] for SDRSZ = 01b (12 row pins), or A[15+n:15] for SDRSZ = 11b (13 row pins); where n = 0 when SDBSZ = 0, and n = 1 when SDBSZ = 1. This ensures that the SDRAM bank select inputs are correct during READ and WRT commands. Thus, the EMIF maintains these values as shown in both row and column addresses.

Table 2–13. VC5501/5502 Byte Address-to-Address Pin (A[21:2]) Mapping for 8-, 16-, and 32-bit SDRAM Interface

			TMS320VC5501/5502 Address															
			A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
Column Address Bits	Data Bus Width	DRAM Cmd	SDRAM Address															
			A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
8	8	$\overline{RAS}$	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
		$\overline{CAS}$	23	21	21	20	19	Low	Low	Low	7	6	5	4	3	2	1	0
	16	$\overline{RAS}$	Low	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9
		$\overline{CAS}$	Low	23	22	21	20	Low	Low	Low	8	7	6	5	4	3	2	1
	32	$\overline{RAS}$	Low	Low	23	22	21	20	19	18	17	16	15	14	13	12	11	10
		$\overline{CAS}$	Low	Low	Low	22	21	Low	Low	Low	9	8	7	6	5	4	3	2
9	8	$\overline{RAS}$	Low	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9
		$\overline{CAS}$	Low	23	22	21	20	Low	Low	8	7	6	5	4	3	2	1	0
	16	$\overline{RAS}$	Low	Low	23	22	21	20	19	18	17	16	15	14	13	12	11	10
		$\overline{CAS}$	Low	Low	23	22	21	Low	Low	9	8	7	6	5	4	3	2	1
	32	$\overline{RAS}$	Low	Low	Low	23	22	21	20	19	18	17	16	15	14	13	12	11
		$\overline{CAS}$	Low	Low	Low	23	22	Low	Low	10	9	8	7	6	5	4	3	2
10	8	$\overline{RAS}$	Low	Low	23	22	21	20	19	18	17	16	15	14	13	12	11	10
		$\overline{CAS}$	Low	Low	23	22	21	Low	9	8	7	6	5	4	3	2	1	0
	16	$\overline{RAS}$	Low	Low	Low	23	22	21	20	19	18	17	16	15	14	13	12	11
		$\overline{CAS}$	Low	Low	Low	23	22	Low	10	9	8	7	6	5	4	3	2	1
	32	$\overline{RAS}$	Low	Low	Low	Low	23	22	21	20	19	18	17	16	15	14	13	12
		$\overline{CAS}$	Low	Low	Low	Low	23	Low	11	10	9	8	7	6	5	4	3	2

2.6.5 SDRAM Refresh

The RFEN bit in the SDRAM control register 1 selects the SDRAM refresh mode of the EMIF. A value of 0 in the RFEN field disables all EMIF refreshes; the user must insure that refreshes are implemented in an external device. A value of 1 in the RFEN field enables the EMIF to perform refreshes of SDRAM.

The refresh command (REFR) is sent to all CE spaces configured to use SDRAM by the MTYPE field of the corresponding CE space control register. REFR is automatically preceded by a DCAB command. This ensures all CE spaces selected with SDRAM are deactivated before refresh occurs. Page information is always invalid before and after a REFR command; thus a refresh cycle always forces a page miss on the next access. Following the DCAB command, the EMIF begins performing “trickle” refreshes at a rate defined by the PERIOD value in the EMIF SDRAM period register, provided no other SDRAM access is pending.

The SDRAM interface monitors the number of refresh requests posted to it and performs them. Within the EMIF SDRAM control block, a 2-bit counter monitors the backlog of refresh requests. The counter increments once for each refresh request and decrements once for each refresh cycle performed. The counter saturates at the value of 11, and also at 00. At reset, the counter is automatically set to 11 to ensure that several refreshes occur before accesses begin.

When EMIF refreshing, the following rules apply:

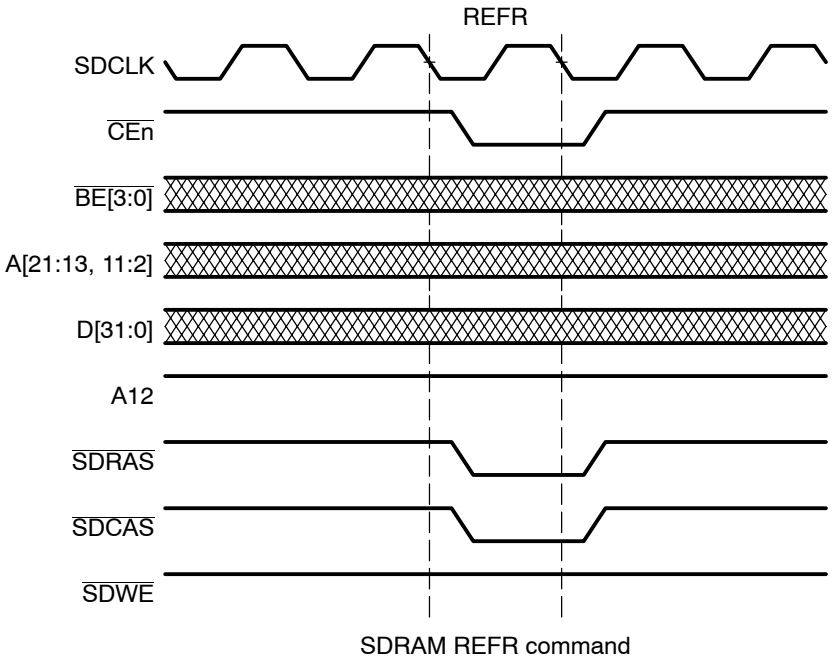
- The value of 11 indicates an urgent refresh condition, causing the page information register to be invalidated and forcing the controller to close the current SDRAM page. Thus, the EMIF SDRAM controller performs three REFR commands, decrementing the counter to 00 following the DCAB command before proceeding with the remainder of the current access. If SDRAM is present in multiple CE spaces, the DCAB-refresh sequence occurs in all spaces containing SDRAM.
- During idle times on the SDRAM interface(s), if no request is pending from the EMIF, the SDRAM interface performs REFR commands as long as the counter value is nonzero. This feature reduces the likelihood of having to perform urgent refreshes during actual SDRAM accesses later. If SDRAM is present in multiple CE spaces, this refresh occurs only if all interfaces are idle with invalid page information.

The C5501/5502 REFR requests are high priority, and no distinction exists between urgent and trickle refresh. Transfers in progress are allowed to complete. The C5501/5502 SDRAM refresh periods are allowed to complete. The C5501/5502 SDRAM refresh period has an extra bit field, XRFR, which controls the number refreshes performed when the counter reaches zero. This feature allows the XRFR field to be set to perform up to four refreshes when the refresh counter expires.

However, during EMIF idle or system idle, the SDRAM clock is shut off to save power. EMIF does not issue any SDRAM refresh command to the SDRAM interface and the data stored in the SDRAM can be lost. It is the user's responsibility to refresh SDRAM, if SDRAM refresh is needed.

The EMIF SDRAM interface perform $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh cycles for SDRAM. Some SDRAM manufacturers call this autorefresh. Prior to a REFR command, a DCAB command is performed to all CE spaces specifying SDRAM to ensure that all active banks are closed. Page information is always invalid before and after a REFR command; thus, a refresh cycle always forces a page miss. A deactivated cycle is required prior to the refresh command. Figure 2−14 shows the timing diagram for an SDRAM refresh.

Figure 2−14. SDRAM Refresh Timing Diagram



2.6.6 SDRAM Self Refresh Mode

The SLFRFR bit of the SDRAM control register (SDCTL) forces the EMIF to place the external SDRAM in a low-power mode, called Self Refresh, in which the SDRAM maintains valid data while consuming a minimal amount of power. This mode is entered when a 1 is written to the SLFRFR bit and SDRAM exists in the system. When the SLFRFR bit is set, the refresh enable bit RFEN in the SDCTL must be written with a 0 simultaneously. When the SLFRFR bit is asserted, all open p ages of SDRAM are closed (DCAB issued to all CE spaces). In addition, a REFRESH command is issued on the same cycle that the SDCKE signal is driven low.

It is the user's responsibility to ensure that the SLFRFR bit is turned on/off at appropriate times. To exit SLFRFR mode, write a 0 to the SLFRFR bit, and then immediately read back before performing other accesses. As long as SLFRFR=1, the user should ensure that no SDRAM accesses are performed.

During self refresh mode, the SDRAM clock (ECLKOUT1) can be turned off if the system does not use the Hold interface, or if ECLKOUT1 is not used elsewhere in the system. ECLKOUT1 must be re-enabled before exiting self refresh mode. The EMIF ensures that the SDRAM is in the self refresh state for at least TRAS cycles, where TRAS is defined in the SDEXT register. In addition, the EMIF ensures the time from SDCKE high to the next ACTV command is at least 16 ECLKOUT1 cycles.

If SDRAM is not in use int he system, the ADCKE pin can be used as a general purpose output. The inverse of SLFRFR bit is driven on the SDCKE pin. If a Hold request is detected, before acknowledging this request with HOLDA the EMIF asserts the SDCKE output (as long as TRAS requirement has been met) and clear the SLFRFR bit to wake the SDRAM from reset. If SDRAM is not in use by the system, HOLD has no effect on the state of the SDCKE output or the SLFRFR field.

The effects of the SLFRFR bit with an SDRAM in the system are summarized here:

- Write to SLFRFR while not in Hold causes self refresh mode entry/exit.
- Write to SLFRFR while in Hold: write to SLFRFR is ignored, bit is not written.
- If HOLD request occurs while SLFRER=1, the EMIF ensures that the device has been in self refresh mode at least TRAS cycles. Then the EMIF exits self refresh mode (deasserts SLFRER). After 16 ECLKOUT1 cycles, the EMIF acknowledges the HOLD request.

Note:

The EMIF SDCKE signal must be connected to the SDRAM CKE signal for proper SLFRER operation.

2.6.7 Mode Register

The mode register is a register located in the external SDRAM memory that dictates its operating characteristics. When initializing SDRAM, the EMIF must set this register to the value described here before normal read or write accesses can occur.

The EMIF automatically performs a DCAB command followed by a MRS command whenever the INIT register is written to. Like DCAB and REFR commands, MRS commands are sent to all CE spaces configured as SDRAM. The EMIF always uses a mode register value of 0x0032 or 0x0022 during a MRS command. Figure 2−15 shows the mapping between mode register bits, EMIF pins, and the mode register value. Table 2−14 shows the JEDEC standard SDRAM configuration values selected by this mode register value. Figure 2−16 shows the timing diagram during execution of the MRS command.

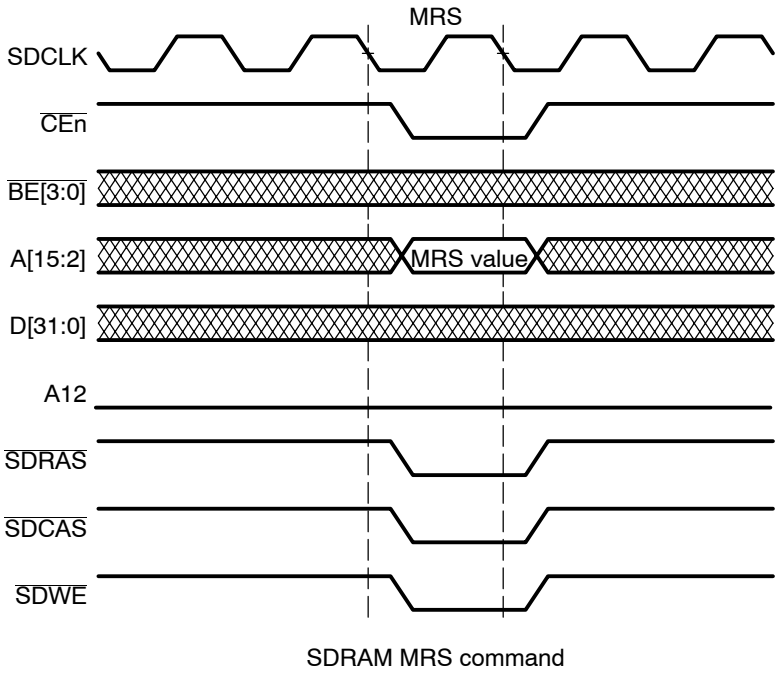
Figure 2−15. Mode Register Value

Mode Register Bit	11	10	9	8	7	6	5	4	3	2	1	0
EMIF Pins	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
Field	Reserved		Write Burst Length	Reserved		Read Latency			S/I	Burst Length		
Value	0	0	0	0	0	0	1	1/0	0	0	0	0

Table 2−14. Implied SDRAM Configuration by MRS Value

Field	CASL=0 (bit 4=0)	CASL=1 (bit 4=1)
Write Burst Length	4 words	4 words
Read Latency	2 cycles	3 cycles
Serial/Interleave Burst Type	Serial	Serial
Burst Length	4 words	4 words

Figure 2−16. SDRAM Module Register Set: MRS Command Timing Diagram



2.6.8 Timing Requirements

SDRAM timing parameters decouple the EMIF from SDRAM speed limitations. Six of these parameters are programmable via the EMIF SDRAM control registers; the remaining parameters are assumed to be static values as shown in . The six programmable values assure that EMIF control of SDRAM obeys these minimum timing requirements. Consult the manufacturer's data sheet for the particular SDRAM.

Table 2–15. SDRAM Timing Parameters

Parameter	Description	Value in CLKMEM Cycles
T_{rc}	REFR command → ACTV/MRS/REFR command	TRC + 1
T_{rcd}	ACTV command → READ/WRT command	TRCD + 1
T_{rp}	DCAB command → ACTV/MRS/REFR command	TRP + 1
T_{ras}	ACTV command → DCAB command	TRAS + 1
T_{mrd}	MRS command → ACTV/DCAB command	MRD + 1
T_{cl}	CAS Latency	TCL + 2
T_{rrd}	ACTV bank A → ACTV bank B (same CE space)	TRRD + 2
T_{wr}	Write recovery, time from last data out of C5501/5502 (write data) to DEAC/DCAB command	TWR + 1
T_{hzp}	High Z from precharge, time from DEAC/DCAB to SDRAM outputs (read data) in high Z	THZP + 1

Table 2−16. C5501/5502 Recommend Values for Command-to-Command Parameters

Parameter	Description	Value in EMIF Clock Cycles	Suggested Value for CL=2	Suggested Value for CL=3
READ to READ	READ command to READ command. Used to interrupt a READ burst for random READ addresses	RD2RD + 1	RD2RD = 0	RD2RD = 0
READ to DEAC	Used in conjunction with t_{HZP}. Specifies the minimum amount of time between READ command and DEAC/DCAB command	RD2DEAC + 1	RD2DEAC = 1	RD2DEAC = 1
READ to WRITE	READ to WRITE command. The value programmed in this parameter depends on t_{CL} . READ to WRITE should be CAS latency plus 2 cycles (in EMIF clock cycles) to provide 1 turnaround cycle before WRITE command.	RD2WR + 1	RD2WR = 3	RD2WR = 4
BEx high before write interrupting read	Specifies the number of cycles that the BEx outputs should be high before a write is allowed to interrupt a read. This is related to READ to WRITE parameter.	R2WDQM + 1	R2WDQM = 1	R2WDQM = 2
WRITE to WRITE	Number of cycles between a WRITE interrupting a WRITE. Used for random WRITEs.	WR2WR + 1	WR2WR = 0	WR2WR = 0
WRITE to DEAC	Number of cycles between a WRITE command and a DEAC/DCAB command	WR2DEAC + 1	WR2DEAC = 1	WR2DEAC = 1
WRITE to READ	Number of cycles between a WRITE command and a READ command	WR2RD + 1	WR2RD = 0	WR2RD = 0

2.6.9 Deactivation

The SDRAM deactivation (DCAB) is performed after a hardware reset or when INIT = 1 in the EMIF SDRAM register. This cycle is also required by the SDRAMs prior to REFR, MRS, and when a page boundary is crossed. During the DCAB command, SDA10 is driven high to ensure that all SDRAM banks are deactivated.

The C5501/5502 also supports the DEAC command, whose operation is depicted in Figure 2−18, which closes a single page of SDRAM specified by the banks selected signals. When page boundary is crossed, the DEAC command is used to close the open page

Figure 2−17 shows the timing diagram for SDRAM deactivation all banks and Figure 2−18 shows the timing diagram for SDRAM deactivation a single bank.

Figure 2−17. SDRAM Deactivation − All Banks

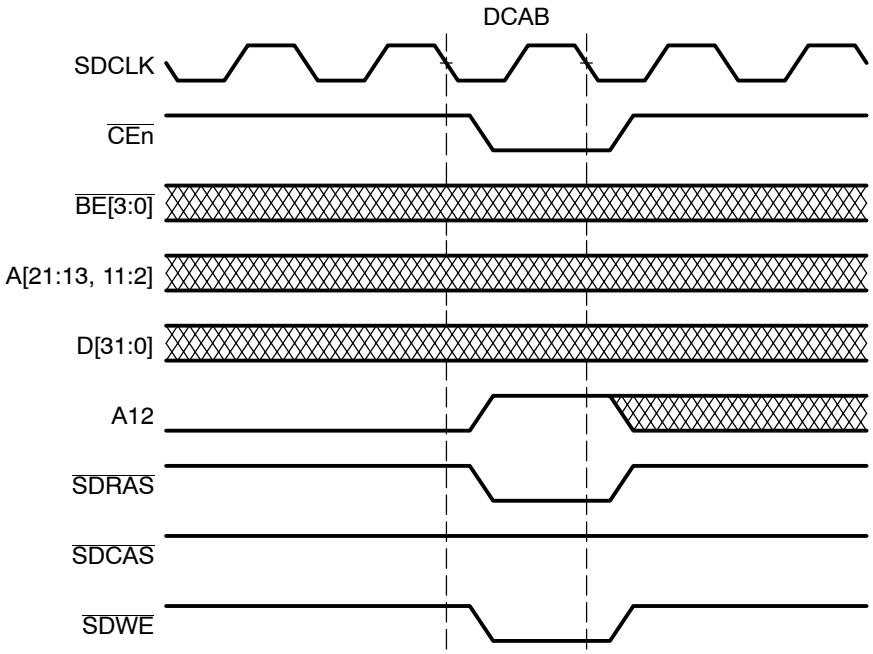
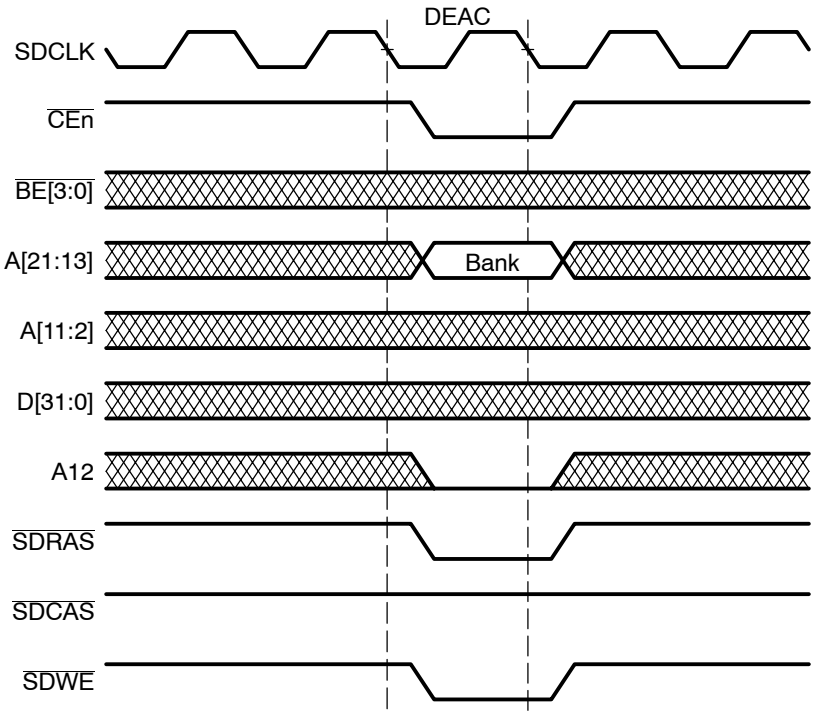


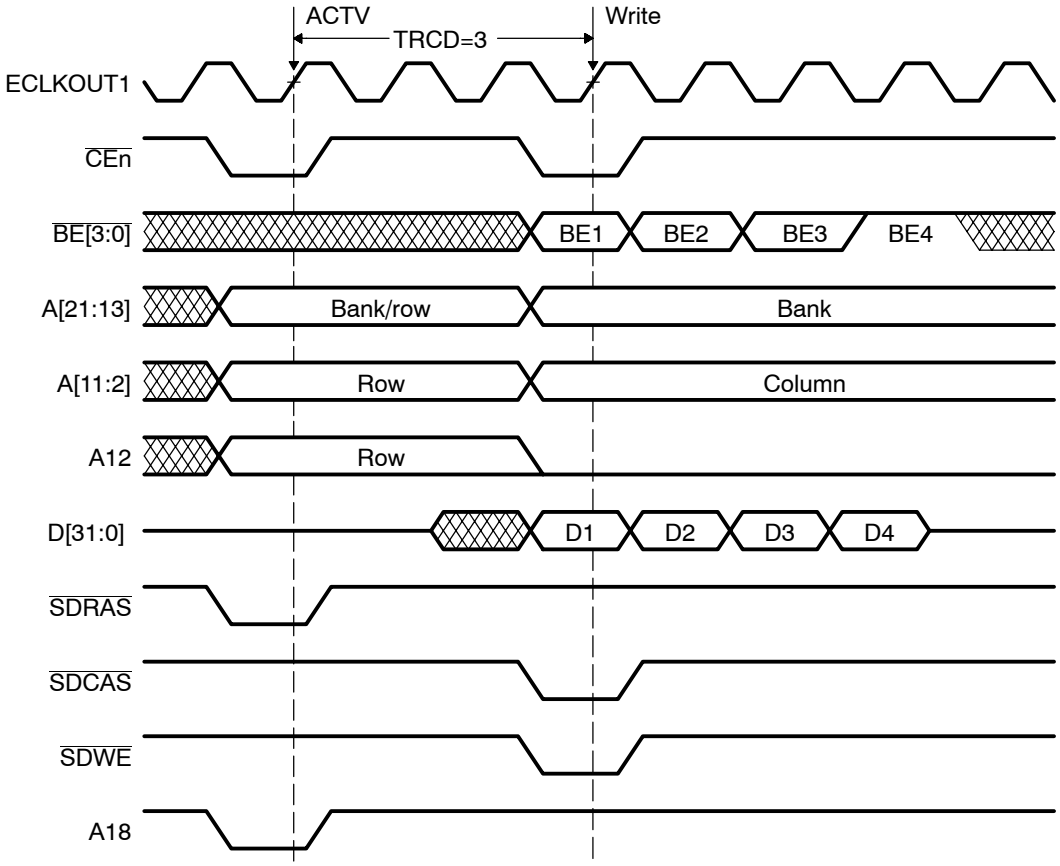
Figure 2−18. SDRAM Deactivation − Single Bank



2.6.10 Activation

The EMIF automatically issues the Activate (ACTV) command before a read or write to a new row of SDRAM. The ACTV command opens up a page of memory, allowing future accesses (reads or writes) with minimum latency. When the ACTV command is issued by the EMIF, a delay of T_{rcd} is incurred before a read or write command is issued. Figure 2−19 shows an example of an ACTV command before an SDRAM write. In this example, t_{RCD} = 3 EMIF clock cycles. The ACTV command for SDRAM reads are identical. Reads or writes to the currently active row and bank of SDRAM can achieve much higher throughput than reads or writes to random areas, because every time a new page is accessed, the ACTV command must be issued.

Figure 2−19. ACTV Command Before an Example SDRAM Write

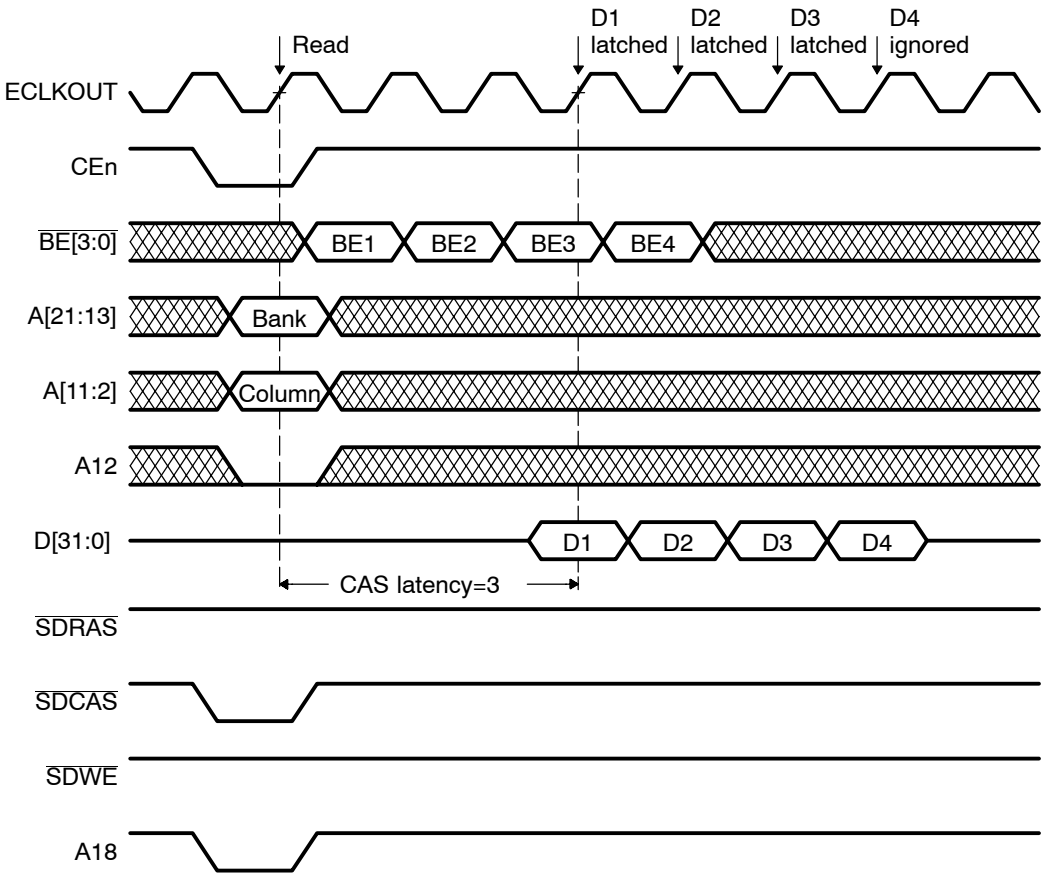


2.6.11 SDRAM Read

During a SDRAM read, the selected bank is activated with the row address during the ACTV command. In this example (which is described in a future document), three read commands are performed to three different column addresses in the same page. The EMIF uses a $\overline{CAS}$ latency of 2 or 3 and a burst length of 4. The 3-cycle read latency causes data to appear 3 cycles after the corresponding column address. Since the default burst length is four words, the SDRAM returns four pieces of data for every read command. If no additional accesses are pending to the EMIF, as in Figure 2−20, the read burst completes and the unneeded data is disregarded. If accesses are pending, the read burst can be interrupted with a new command (READ, WRT, DEAC, DCAB), controlled by the SDRAM extension register. If a new access is not pending, the DCAB/DEAC command is not performed until the page information becomes invalid.

If a refresh cycle or an access to a different page of memory is required, then following the last column access, a DCAB cycle is performed to deactivate the bank. An idle cycle is inserted between the final read command and the DCAB command to meet SDRAM timing requirements. Note that the transfer of data completes during and past the DCAB command. If no new access is pending, the DCAB command is not performed until such time that the page information becomes invalid.

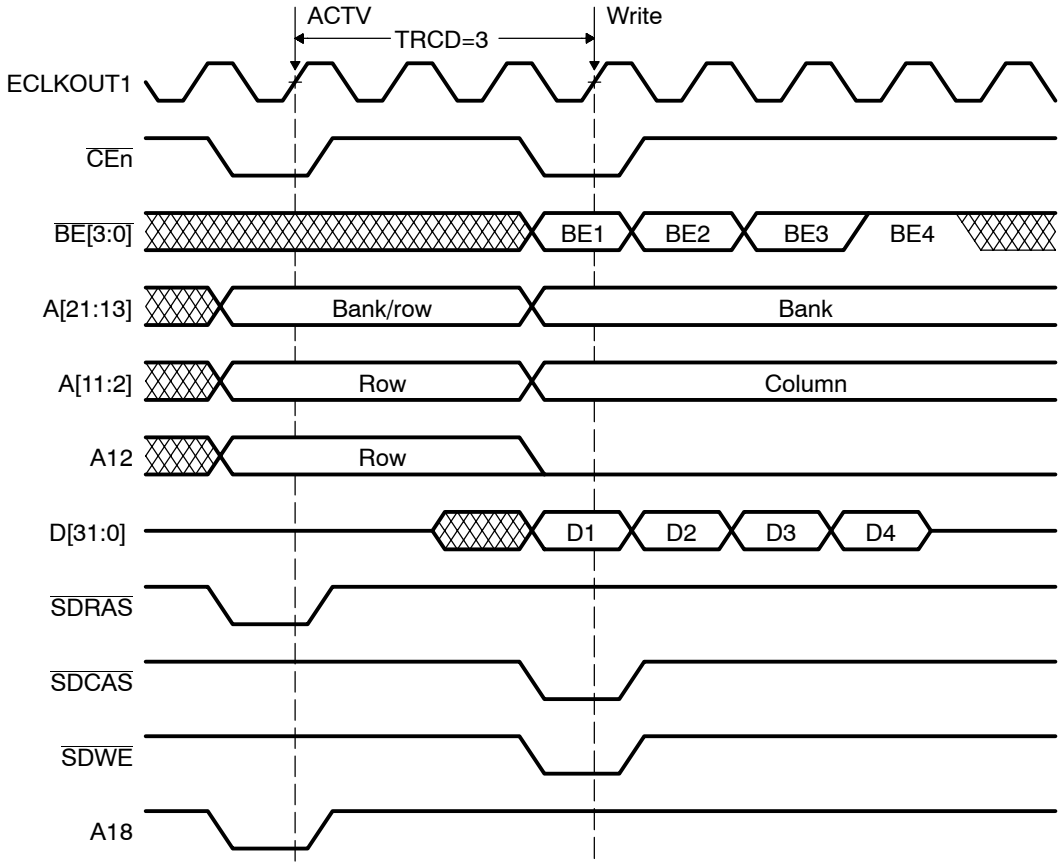
Figure 2–20. TMS320C5501/5502 SDRAM Read



2.6.12 SDRAM Write

All SDRAM writes have a burst length. The bank is activated with the row address during the ACTV command. There is no latency on writes, so data is output on the same cycle as the column address. Writes to particular bytes are disabled via the appropriate DQM inputs; this feature allows for byte and half-word writes. Figure 2–21 shows the timing for a three-word write on the C5501/5502. Since the default C5501/5502 write-burst length is four words, the last write is masked out via the byte enable signals. On the C5501/5502, idle cycles are inserted as controlled by the parameters of the SDRAM extension register fields (WR2RD, WR2DEAC, WR2WR, TWR). The bank is then deactivated with a DEAC command for C5501/5502, and the memory interface can begin a new page access. If no new access is pending, the DEAC command is not performed until the page information becomes invalid. The values on the bank select bits during column accesses and during the DEAC command are the values latched during the ACTV command.

Figure 2–21. C5501/5502 SDRAM Write



2.7 Programmable Synchronous Interface

The C5501/5502 EMIF offers additional flexibility by replacing the SBSRAM interface with a programmable synchronous interface. The programmable synchronous interface supports glueless interfaces to the following devices:

- ❏ Pipelined and flow-through SBSRAM
- ❏ Zero bus turnaround (ZBT) synchronous pipeline SRAM
- ❏ 32-bit-wide synchronous FIFOs in standard and first word fall through (FWFT) mode

The programmable synchronous interface is configured by the CE space secondary control register (CExSEC). The bit fields in the CExSEC control the cycle timing, and the clock used for programmable synchronous interface synchronization. See section 2.12.2 for a detailed description of the bit fields.

A new command is issued every cycle for SBSRAM bursts, and a deselect cycle is issued at the end of the burst. The RENEN field in the CExSEC register should be set to 0 for the SBSRAM interface to enable the SADS signal. See section 2.5 for details on the SBSRAM interface.

Table 2−17 shows the programmable synchronous interface pins.

Table 2−17. TMS320C5501/5502 Programmable Synchronous Pins

EMIF Signal	Signal Function
$\overline{\text{SADS}}$/$\overline{\text{SRE}}$	Address strobe/Read enable (selected by RENEN)
$\overline{\text{SOE}}$	Output enable
$\overline{\text{SOE3}}$	Output enable for CE3. The SOE3 pin is not multiplexed with other signals (useful for glueless FIFO interface)
$\overline{\text{SWE}}$	Write Enable
ECLKOUT1	Synchronous interface clock, runs at 1x EMIF input clock rate
ECLKOUT2	Synchronous interface clock, runs at 1x, 1/2x, or 1/4x EMIF input clock rate

2.7.1 ZBT SRAM Interface

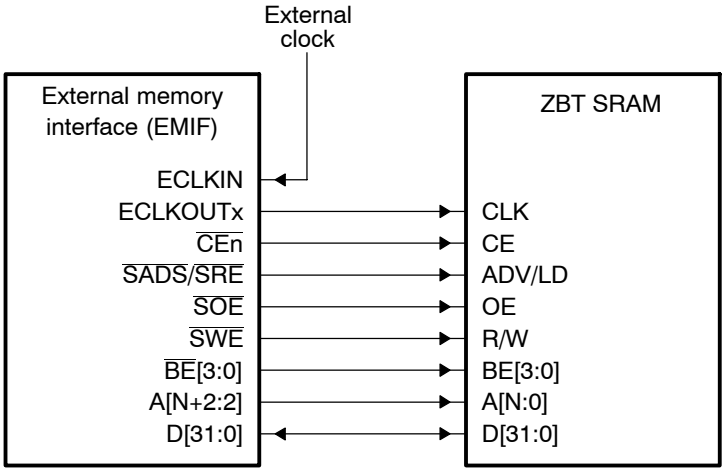
The programmable synchronous mode supports zero bus turnaround (ZBT) SRAM interface.

For ZBT SRAM interface, the following fields in the CExSEC register must be set as:

- SYNCRL = 10b; 2 cycle read latency
- SYNCWL = 10b; 2 cycle write latency
- CEEXT = 0; CE goes inactive after the final command has been issued
- RENEN = 0; SADS/SRE signal acts as SADS signal.

Figure 2−22 shows the ZBT SRAM interface.

Figure 2−22. TMS320C5501/5502 ZBT SRAM Interface



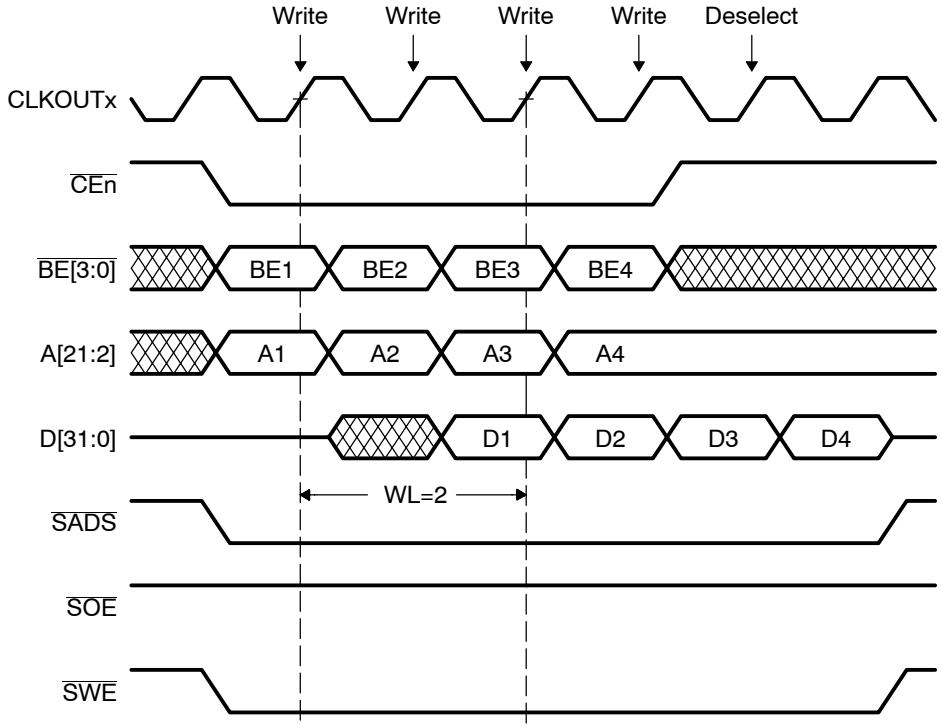
2.7.1.1 ZBT SRAM Read

ZBT SRAM read waveforms are identical to the SBSRAM read waveforms, since the register settings corresponding to the reads are the same. Refer to section 2.5.2 for details.

2.7.1.2 ZBT SRAM Write

For ZBT SRAM writes, the control signal waveforms are exactly the same as standard SRAM writes. The write data, however, is delayed by two cycles, as controlled by SYNCWL = 10b. Figure 2−23 shows the ZBT SRAM write timing.

Figure 2−23. TMS320C5501/5502 ZBT SRAM Four-Element Write



2.7.2 Synchronous FIFO Interface

The programmable synchronous mode supports both standard timing synchronous FIFO interface, and first word fall through (FWFT) FIFO interface. Since the EMIF always reads 32-bits from external memory, regardless of the original request, only 32-bit-wide synchronous FIFOs are supported.

For synchronous FIFO interface, the following field in the CExSEC register must be set as stated:

RENEN = 1; $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ signal acts as $\overline{\text{SRE}}$ signal

Figure 2−24, Figure 2−25, and Figure 2−26 show the synchronous FIFO interface with glue, the glueless synchronous FIFO interface at CE3 space, and using the dedicated $\overline{\text{SOE3}}$ pin.

Figure 2−24. TMS320C5501/5502 Read and Write Synchronous FIFO Interface With Glue

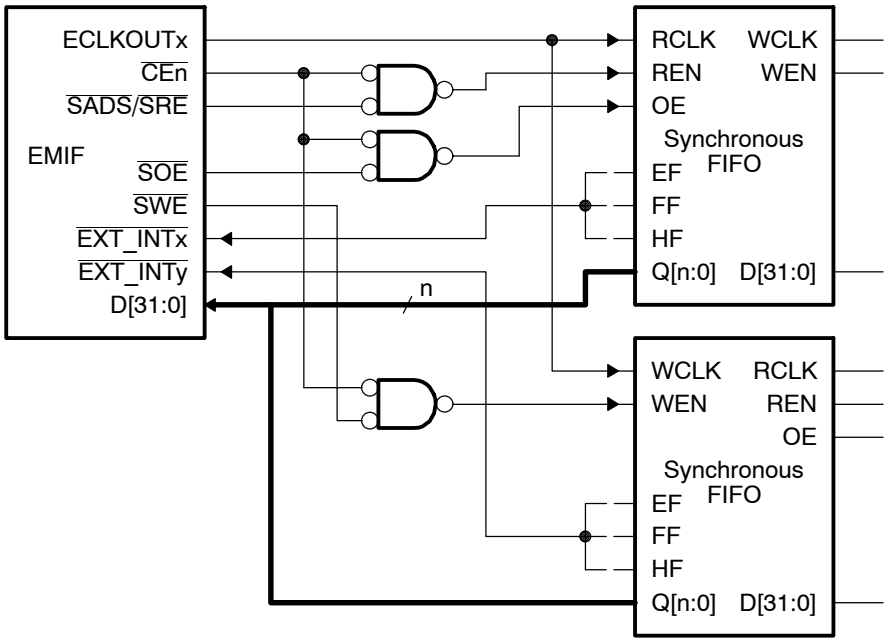


Figure 2−25. TMS320C5501/5502 Glueless Synchronous FIFO Read Interface in CE3 Space

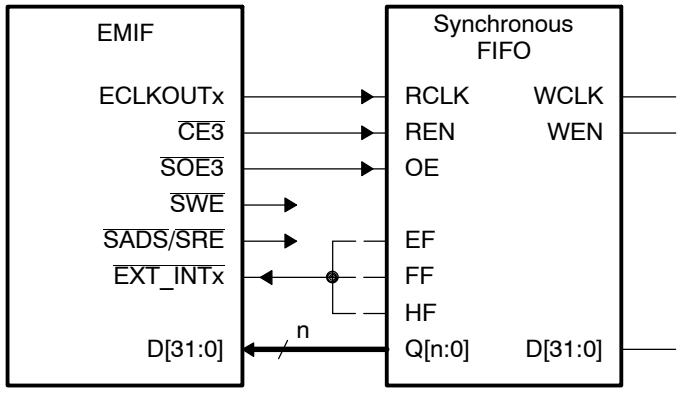
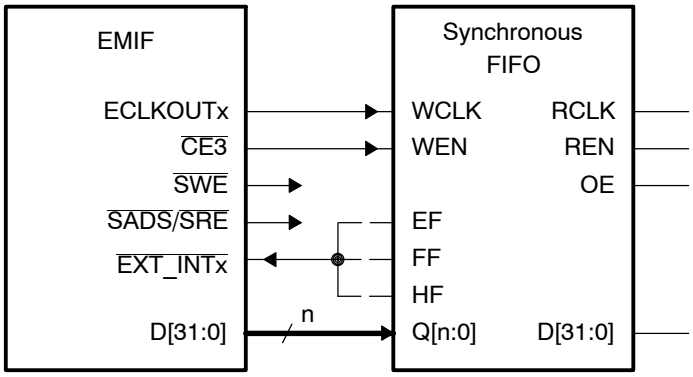


Figure 2−26. TMS320C5501/5502 Glueless Synchronous FIFO Write Interface in CE3 Space

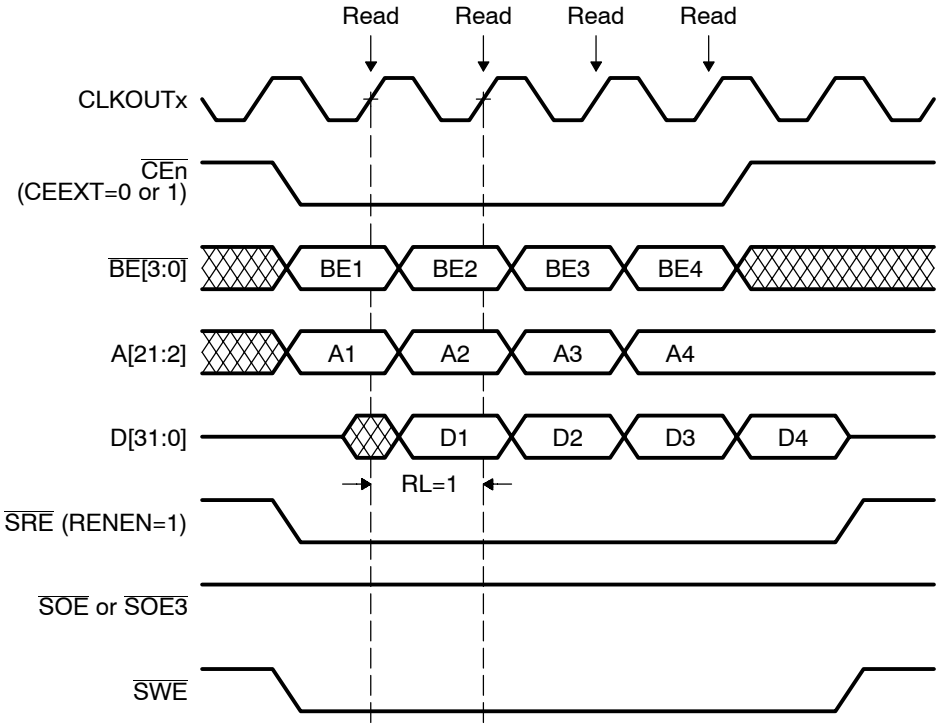


2.7.2.1 Standard Synchronous FIFO Read

Figure 2−27 shows a read of four 32-bit words from a standard synchronous FIFO. The CExSEC register settings are as follows:

- SYNCRL = 01b; one cycle read latency
- RENEN = 1; SADS/SRE signal acts as SRE signal
- CEEXT = 0; used for glueless FIFO interface
  = 1; used for FIFO interface with glue

Figure 2−27. TMS320C5501/5502 Standard Synchronous FIFO Read

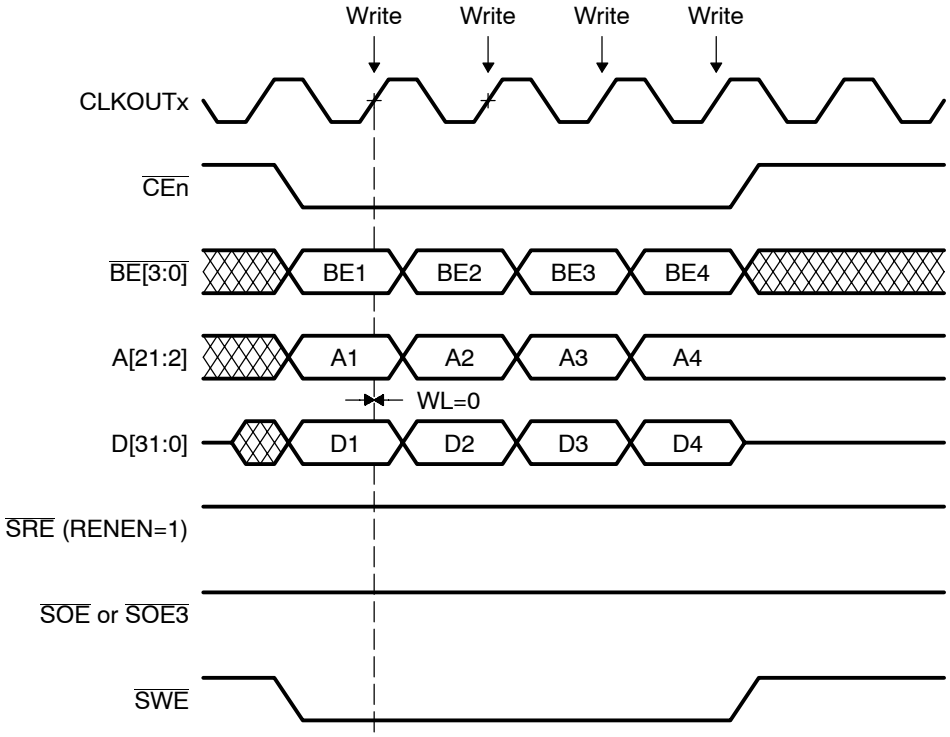


2.7.2.2 Standard Synchronous FIFO Write

Figure 2−28 shows a write of four 32-bit words to a standard synchronous FIFO. The CExSEC register settings are as follows:

- SYNCWL = 00b; zero cycle write latency
- RENEN = 1; $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ signal acts as $\overline{\text{SRE}}$ signal

Figure 2−28. TMS320C5501/5502 Standard Synchronous FIFO Write



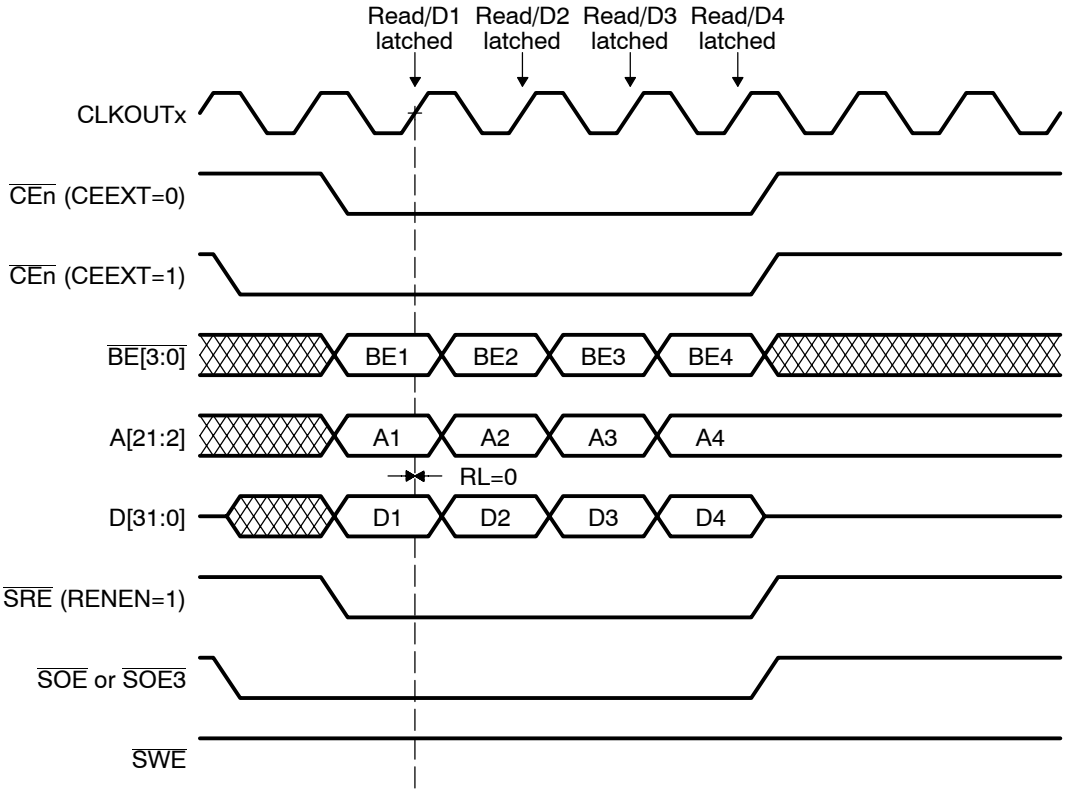
2.7.2.3 FWFT Synchronous FIFO Read

Figure 2−29 shows a read of four 32-bit words from a FWFT synchronous FIFO. The CExSEC register settings are as follows:

- SYNCRL = 00b; zero cycle read latency
- RENEN = 1; $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ signal acts as $\overline{\text{SRE}}$ signal
- CEEXT = 0; used for glueless FIFO interface
  = 1; used for FIFO interface with glue

SYNCRL = 0 causes the $\overline{\text{SOE}}$ (or $\overline{\text{SOE3}}$) signal go active a cycle before the read command begins. If CEEXT = 1, the $\overline{\text{CE}}$ signal goes active at the same time as the $\overline{\text{SOE}}$ signal.

Figure 2−29. TMS320C5501/5502 FWFT Synchronous FIFO Read



2.7.2.4 FWFT Synchronous FIFO Write

The FWFT Synchronous FIFO Write timing is identical to the standard synchronous FIFO write timing. See section 2.7.2.2.

2.8 Program Accesses

When fetching instruction code from external memory, the CPU sends a 32-bit access request to the instruction cache using its program-read data bus (P bus). If the instruction cache is disabled, the request goes directly to the IPORT and then to the EMIF. The EMIF must read 32 bits from the external memory and then pass all 32 bits to the IPORT, which in turn sends the data to the CPU.

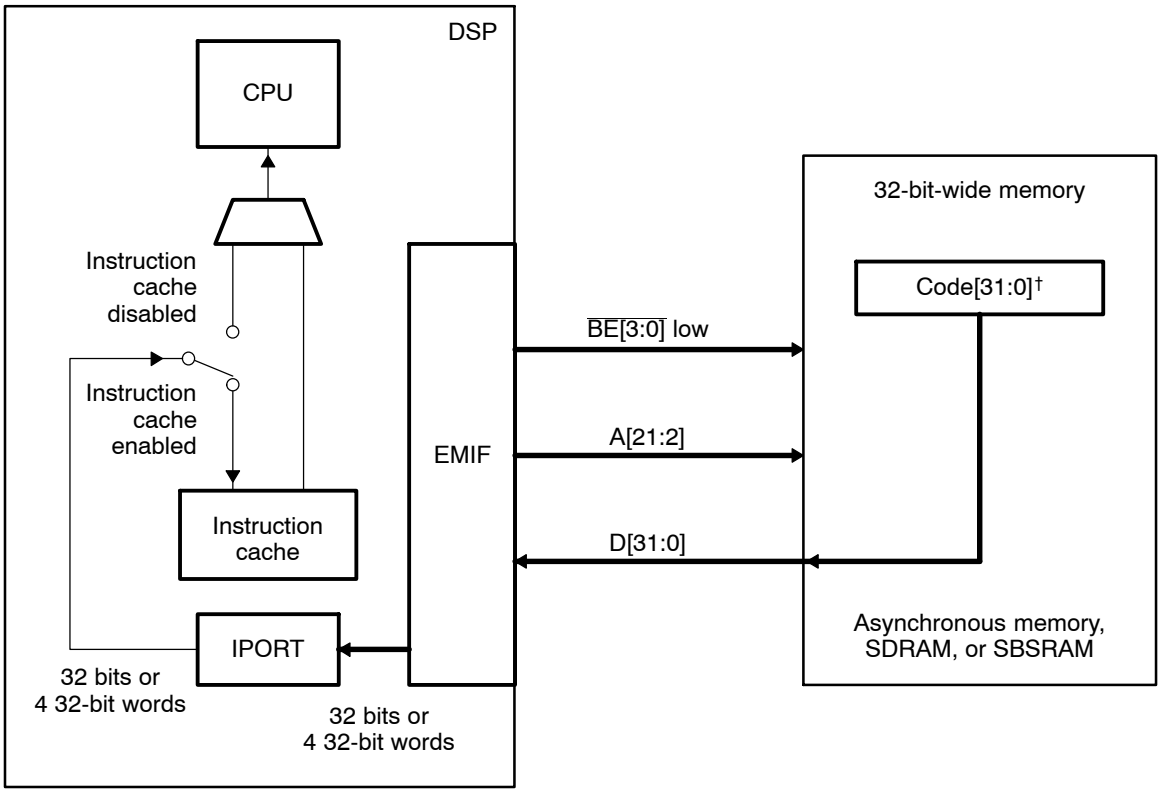
Two things could happen if the instruction cache is enabled. In the case of a cache hit, the CPU request will be immediately serviced by the instruction cache and no data will be read from external memory. In the case of a cache miss, the instruction cache will request four 32-bit words from the EMIF through the IPORT. The EMIF will read four 32-bit words from external memory and then pass the data to the instruction cache through the IPORT. The instruction cache will then send the requested data to the CPU and update its memory contents. More information on the instruction cache can be found in the *TMS320VC5501/5502 DSP Instruction Cache Reference Guide* (SPRU630).

The EMIF can manage the 32-bit access for three memory widths: 32 bits, 16 bits, and 8 bits.

2.8.1 Program Access of 32-Bit-Wide Memory

Figure 2–30 shows how the EMIF behaves when reading program code from 32-bit-wide external memory. The least significant line of the external address bus that is required by 32-bit-wide memory is A2. The external address lines A[21:2] of the EMIF correspond to bits 21-2 of the internal program address. The whole external data bus, D[31:0], is used to transport the data from external memory to the DSP. During an 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 IPORT, which then passes the data to the instruction cache or the P bus as described earlier in this section. The EMIF will read four 32-bit words in the case of a cache miss or one 32-bit word if the instruction cache is disabled.

Figure 2–30. Program Access of 32-Bit-Wide External Memory



† Four 32-bit words are read from external memory in the case of a cache miss.

2.8.2 Program Access of 16-Bit-Wide Memory

Figure 2–31 illustrates the data transfers involved in a program access of 16-bit-wide external memory. The EMIF places a word address on address lines A[21:2]. The 32-bit access is performed as two 16-bit transfers, in two consecutive cycles. During the second cycle, the EMIF automatically increments the first address by 1 to create the second address value.

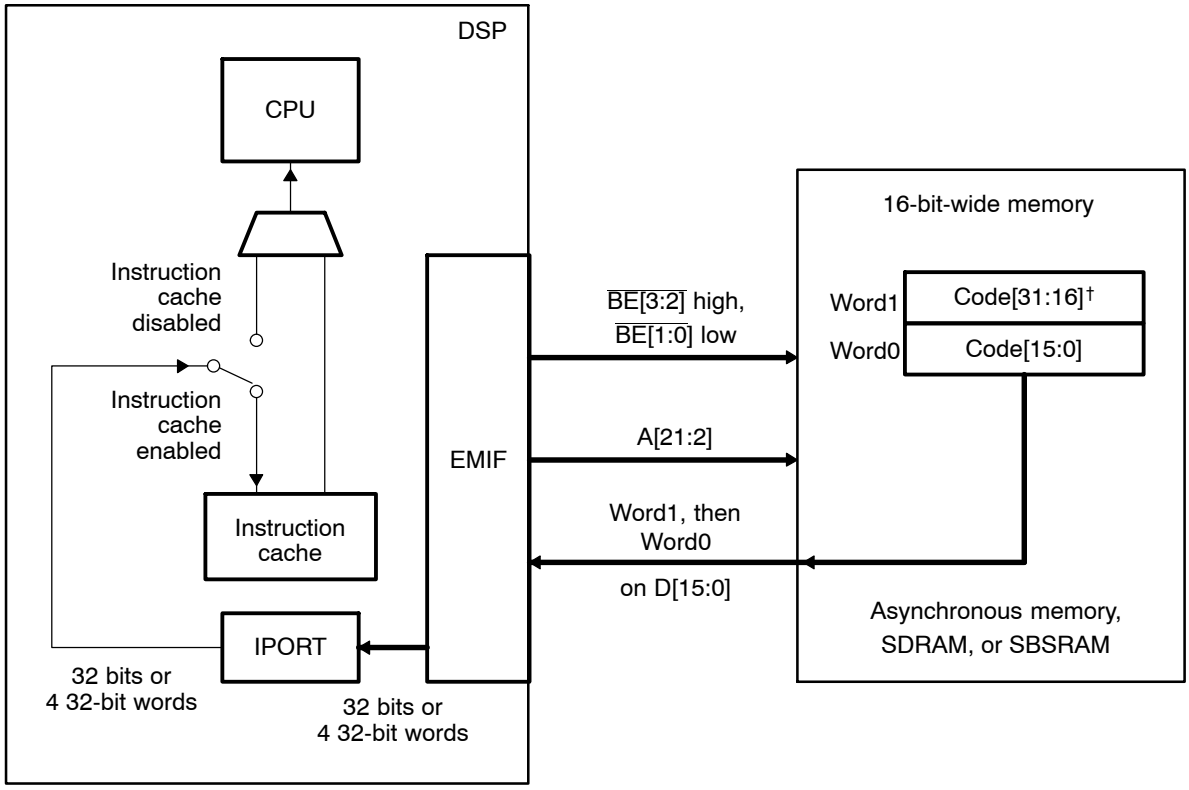
For both 16-bit accesses, the EMIF uses data lines D[15:0]. The 32-bit code block is transferred in the following manner:

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 an access, $\overline{BE3}$ and $\overline{BE2}$ stay high (inactive), and $\overline{BE1}$ and $\overline{BE0}$ are driven low. After an access, the EMIF passes all 32 bits of the code to the IPORT, which then passes the data to the instruction cache or the P bus as described earlier in this section. The EMIF will read eight 16-bit words in the case of a cache miss or two 16-bit words if the instruction cache is disabled.

Figure 2–31. Program Access of 16-Bit-Wide External Memory



† A total of eight 16-bit words are read from external memory in the case of a cache miss.

2.8.3 Program Access of 8-Bit-Wide Memory

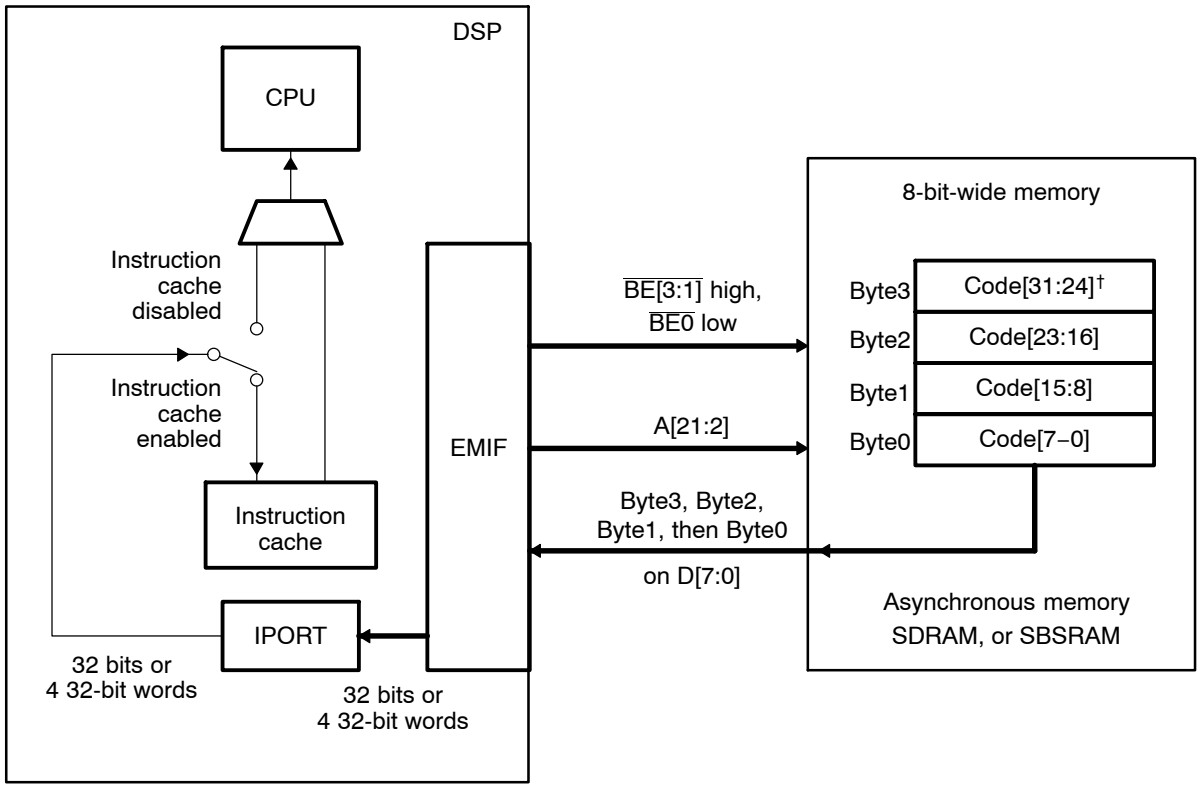
As shown in Figure 2–32, when the EMIF performs a program access of 8-bit-wide memory, the EMIF places a byte address on address lines A[21:2]. The 32-bit access is performed as four 8-bit transfers, in four consecutive cycles. During the second, third, and fourth cycles, the EMIF automatically generates a new address by incrementing the previous address by 1.

For all four 8-bit accesses, 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 an access, $\overline{BE3}$, $\overline{BE2}$, and $\overline{BE1}$ stay high (inactive), and $\overline{BE0}$ is driven low. After an access, the EMIF passes all 32 bits of the code to the IPORT, which then passes the data to the instruction cache or the P bus as described earlier in this section. The EMIF will read 16 bytes in the case of a cache miss or four bytes if the instruction cache is disabled.

Figure 2–32. Program Access of 8-Bit-Wide External Memory



† A total of 16 bytes are read from external memory in the case of a cache miss.

2.9 Data Accesses

The EMIF supports data accesses for:

- 32-bit data (see section 2.9.1)
- 16-bit data (see section 2.9.2 on page 2-61)
- 8-bit data (see section 2.9.3 on page 2-69)

2.9.1 32-Bit Data Accesses

A 32-bit data access is generated by a CPU instruction or a DMA controller operation that reads or writes a 32-bit value. The CPU uses the DPORT to communicate with the EMIF. For each 32-bit read or write operation from the CPU, the DPORT must communicate with two 16-bit CPU buses. For read operations, the C and D buses carry the data from the DPORT to the CPU. For write operations, the E and F buses carry the data from the CPU to the DPORT. The External Memory Port of the DMA controller can place read and write requests directly to the EMIF. The EMIF can manage the 32-bit access for three memory widths: 32 bits, 16 bits, and 8 bits.

2.9.1.1 32-Bit Data Access of 32-Bit-Wide Memory

The process for accessing 32-bit data from 32-bit-wide external memory is illustrated in the two parts of Figure 2−33. The external address lines A[21:2] of the EMIF correspond to bits 21-2 of the internal data address. 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) during an access.

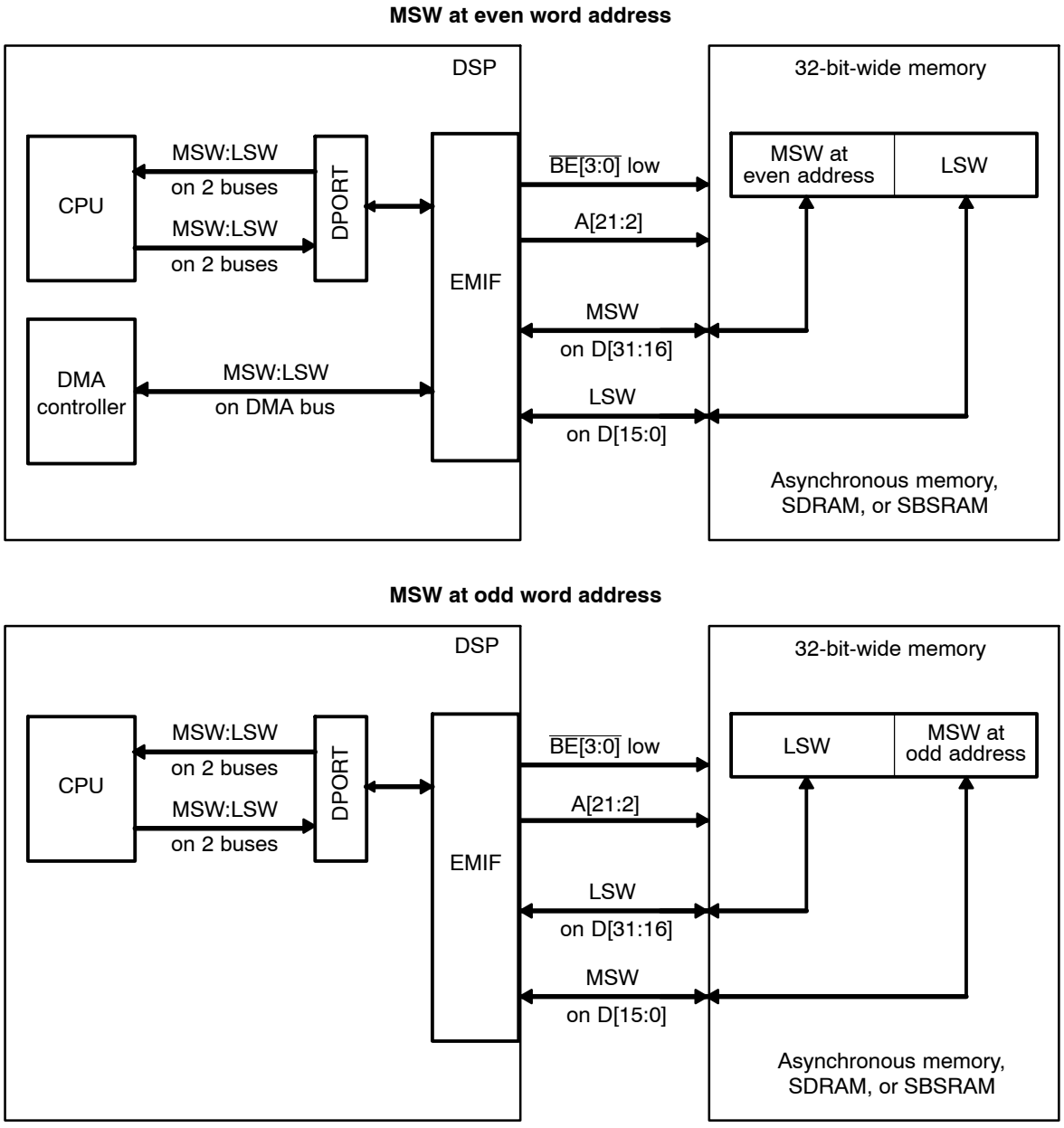
Although only bits 21-2 of the internal address are used by the external memory chip, the EMIF uses bit 1 to determine the positions of the most significant word (MSW) and least significant word (LSW) of the 32-bit value (see Table 2−18). The relative positions of the MSW and LSW determine how the EMIF 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 to make a 32-bit data access, the MSW must be at an even address.

Table 2−18. 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 2−33. 32-Bit Data Access of 32-Bit-Wide External Memory



2.9.1.2 32-Bit Data Access of 16-Bit-Wide Memory

The two parts of Figure 2−34 illustrate the data transfers involved in a 32-bit data access of 16-bit-wide external memory. The external address lines A[21:2] of the EMIF correspond to bits 20-1 of the internal data address. 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 $\overline{BE2}$ stay high (inactive), and $\overline{BE1}$ and $\overline{BE0}$ are driven low.

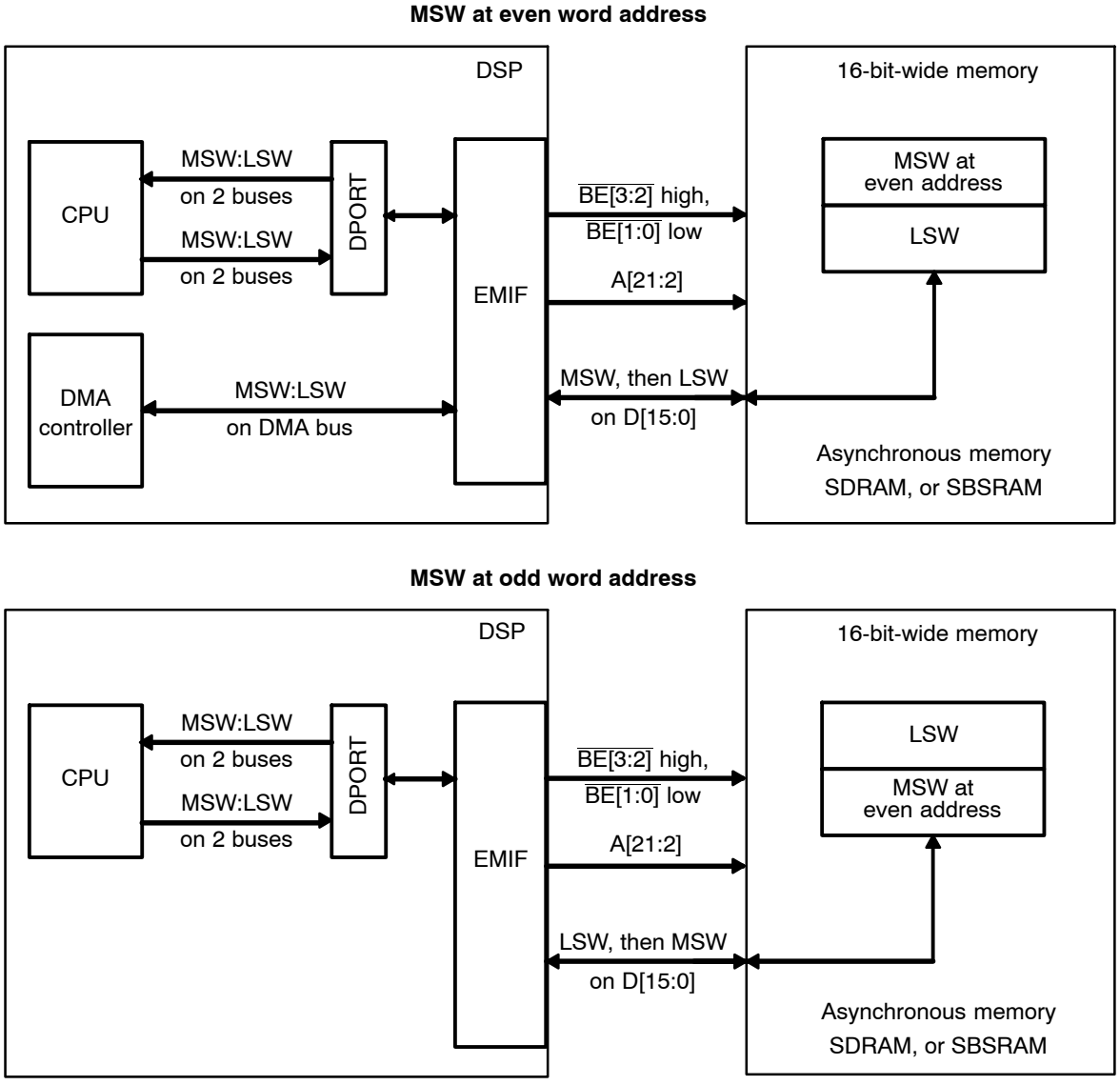
The 32-bit access is performed as two 16-bit transfers across data bus lines D[15:0]. The transfers are performed in two consecutive cycles. During the second cycle, the EMIF automatically generates the second address as described in Table 2−19.

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

Table 2−19. 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	Memory Access
0	The MSW is at an even word address, and the LSW is at the following odd word address.	The EMIF accesses the MSW at the even word address and then the LSW at the next word address.
1	The MSW is at an odd word address, and the LSW is at the previous even word address.	The EMIF accesses the LSW at the even word address and then the MSW at the next word address.

Figure 2−34. 32-Bit Data Access of 16-Bit-Wide External Memory



2.9.1.3 32-Bit Data Access of 8-Bit-Wide Memory

The two parts of Figure 2−35 illustrate the data transfers involved in a 32-bit data access of 8-bit-wide external memory. The external address lines A[21:2] of the EMIF correspond to bits 19-0 of the internal data address. Data bus lines D[7:0] are used to transport the data between the DSP and the external memory. During an access, $\overline{BE3}$, $\overline{BE2}$, and $\overline{BE1}$ stay high (inactive), and $\overline{BE0}$ is driven low.

The 32-bit access is performed as four consecutive 8-bit transfers across data bus lines D[7:0]. The EMIF accesses the 32-bit data as described in Table 2−20.

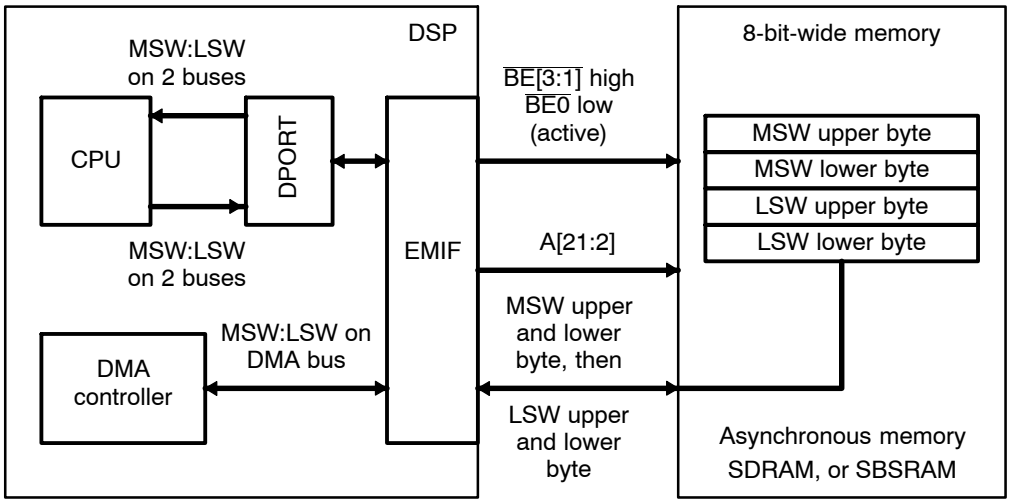
If the DMA controller is to make a 32-bit data access, the MSW must be at an even address.

Table 2−20. The Role of Internal Address Bit 1 During a 32-Bit Data Access of 8-Bit-Wide External Memory

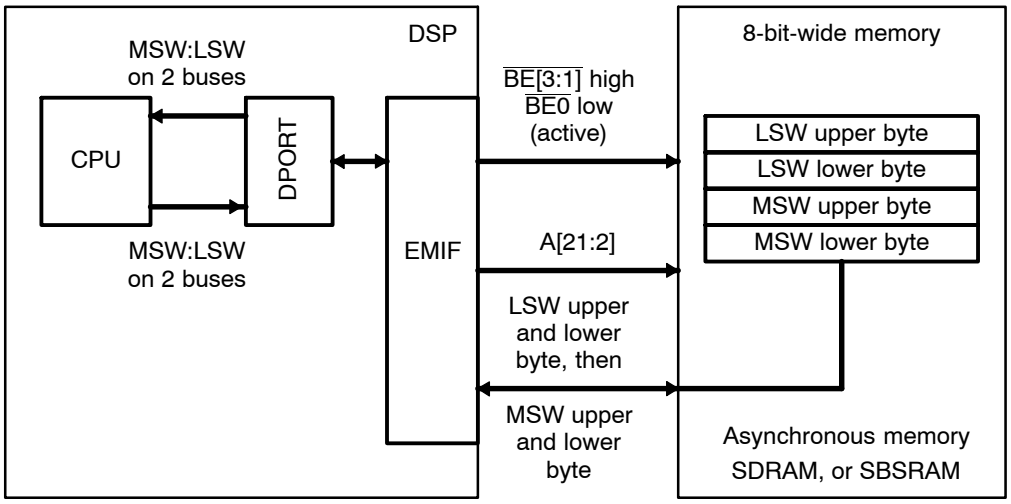
Internal Address Bit 1	External memory is accessed as:
0	MSW is accessed on first two byte accesses, LSW is loaded on the next two byte accesses.
1	LSW is accessed on first two byte accesses, MSW is loaded on the next two byte accesses.

Figure 2−35. 32-Bit Data Access of 8-Bit-Wide External Memory

MSW at even word address



MSW at odd word address



2.9.2 16-Bit Data Accesses

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 manages the 16-bit access for three memory widths: 32 bits, 16 bits, and 8 bits.

Note:

For 16-bit data read accesses, the EMIF will always reads 32-bits from external memory and isolates the desired word in the DSP. For 16-bit data write accesses, the EMIF modifies an individual word.

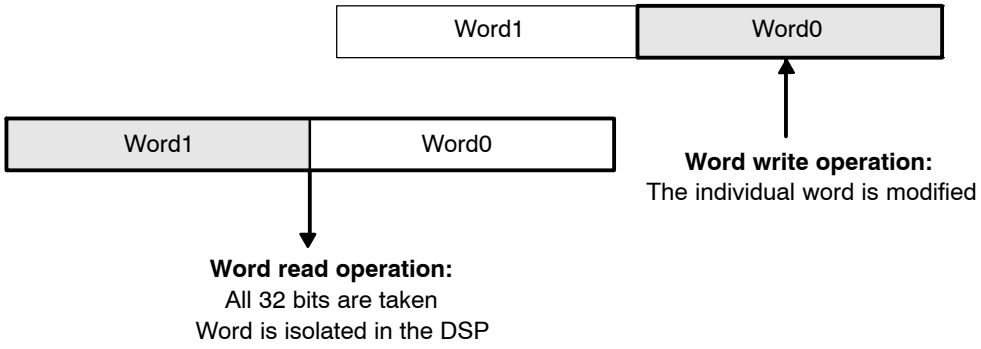
When doing a 16-bit data read, data is eventually discarded as the EMIF always reads 32-bits from external memory. This could significantly lower the overall system throughput because clock cycles are unnecessarily taken up to read unwanted data. However, when the DMA is used to read 16-bit data from external memory, data packing can be used to convert two adjacent 16-bit data read accesses into a single 32-bit data access. As described in section 2.9.1.2, no data is discarded when 32-bit data read accesses are done from 16-bit-wide external memory. More information on DMA data packing can be found in the *TMS320VC5501/5502 DSP Direct Memory Access (DMA) Controller Reference Guide (SPRU613)*.

A similar approach to the DMA can be considered when using the CPU to read 16-bit data from external memory. Instead of doing consecutive 16-bit data accesses, the CPU could do 32-bit data accesses and parse the data internally. Depending on the EMIF settings, parsing could take a smaller number of clock cycles when compared to the number of clock cycles required to read an extra 16-bit data value from external memory.

2.9.2.1 16-Bit Data Access of 32-Bit-Wide Memory

As shown in Figure 2−36, when the EMIF makes 16-bit accesses in 32-bit-wide external memory, the actual width of the access is different for read operations and write operations. When a word is written to external memory, the EMIF modifies an individual word. However, when the EMIF reads a word from external memory, the EMIF reads the full width of the memory, and the desired word is isolated in the DSP.

Figure 2−36. Accessing 16-Bit Data in 32-Bit-Wide Memory



The process for reading 16-bit data from 32-bit-wide external memory is illustrated in the two parts of Figure 2−37. Figure 2−38 illustrates the process for writing 16-bit data to 32-bit wide external memory. The external address lines A[21:2] of the EMIF correspond to bits 21-2 of the internal data address. 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 2−21).

Table 2−21. The Role of Internal Address Bit 1 During a 16-Bit Data Access of 32-Bit-Wide External Memory

Internal Address Bit 1	Operation	Word Is At an …	Data Lines Used	Byte Enable Signal Levels
0	Read	Even word address	D[31:0], bits 31:16 are returned	$\overline{\text{BE}}$[3:0] low (active)
1	Read	Odd word address	D[31:0], bits 15:0 are returned	$\overline{\text{BE}}$[3:0] low (active)
0	Write	Even word address	D[31:16]	$\overline{\text{BE}}$[3:2] low, $\overline{\text{BE}}$[1:0] high (active)
1	Write	Odd word address	D[15:0]	$\overline{\text{BE}}$[3:2] high, $\overline{\text{BE}}$[1:0] low (active)

Figure 2–37. 16-Bit Data Read Access of 32-Bit-Wide External Memory

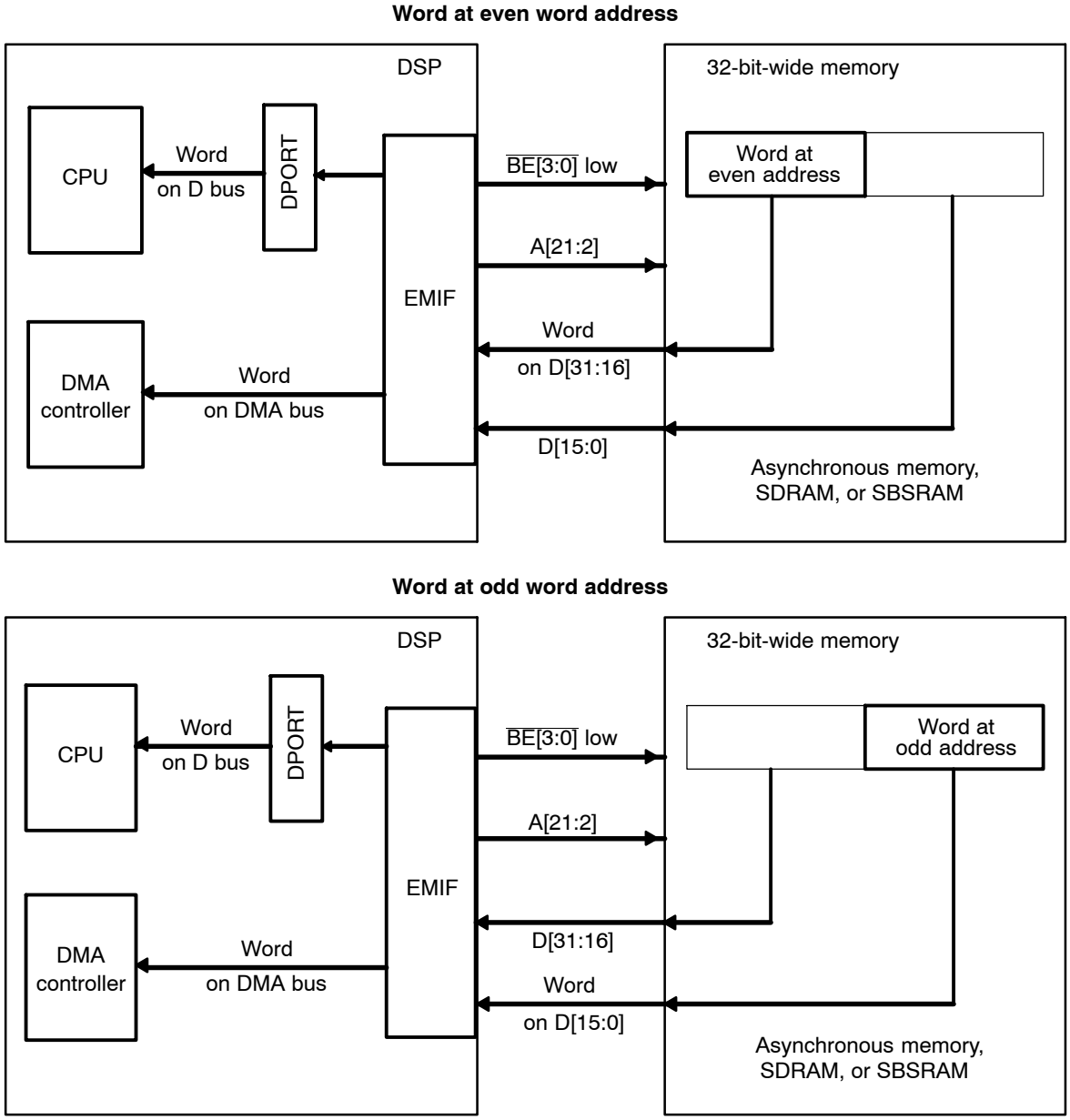
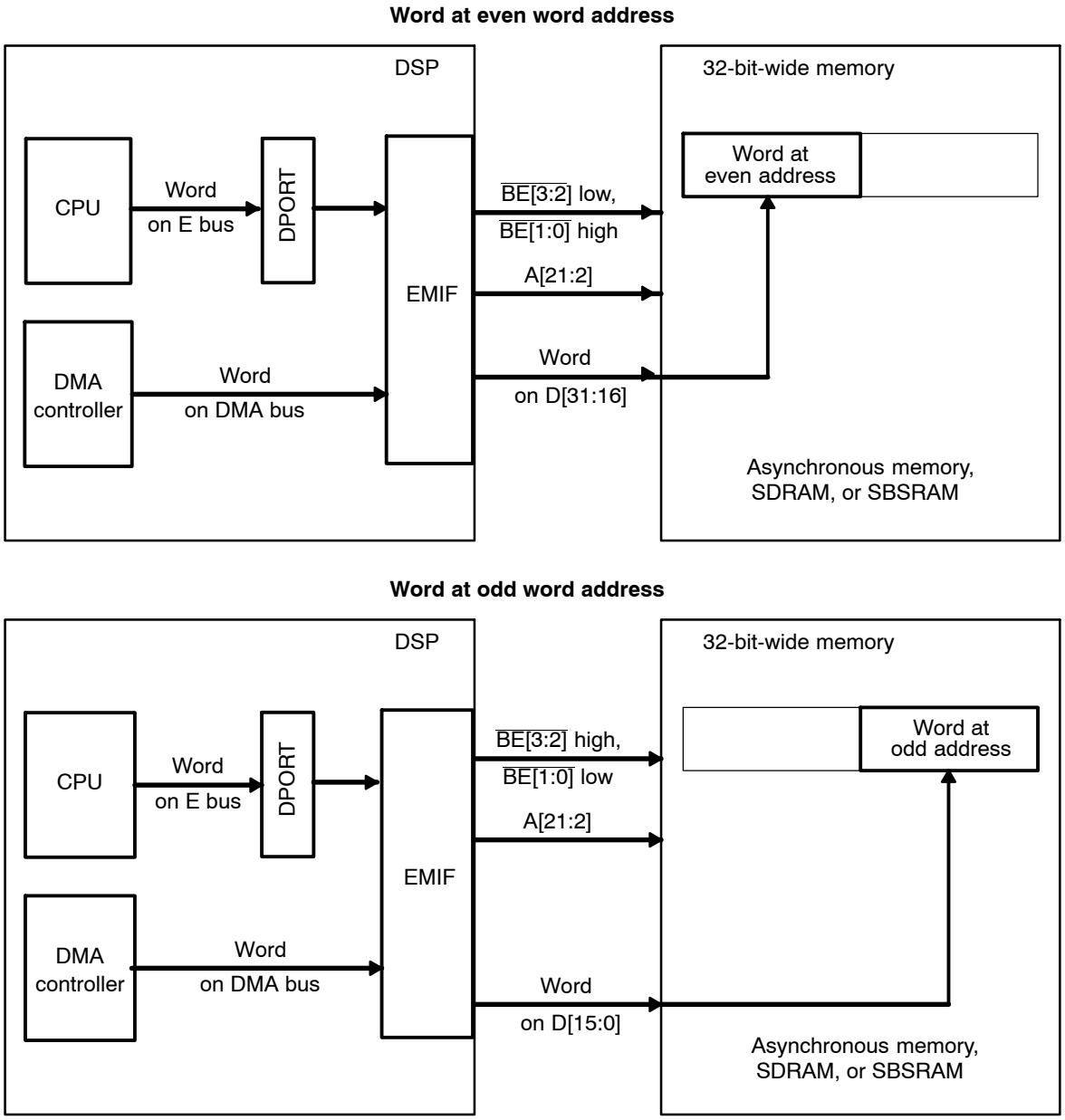


Figure 2−38. 16-Bit Data Write Access of 32-Bit-Wide External Memory



2.9.2.2 16-Bit Data Access of 16-Bit-Wide Memory

Figure 2–39 illustrates the data transfers involved in a 16-bit data write access of 16-bit-wide external memory. A 16-bit data read access from 16-bit-wide external memory is shown on Figure 2–40. The external address lines A[21:2] of the EMIF correspond to bits 20:1 of the internal data address. 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 $\overline{BE2}$ stay high (inactive), and $\overline{BE1}$ and $\overline{BE0}$ are driven low.

When the EMIF makes 16-bit accesses in 16-bit-wide external memory, the number of word accesses for read operations and write operations is different. When a word is written to external memory, the EMIF writes an individual word. However, when a word is read from external memory, the EMIF reads two 16-bit words from external memory (a total of 32-bits), and the desired word is isolated in the DSP. The EMIF generates the addresses for the first and second read cycles as shown in Table 2–22.

Table 2–22. The Role of Internal Address Bit 1 During a 16-Bit Data Read Access of 16-Bit-Wide External Memory

Internal Address Bit 1	For the first read cycle:	For the second read cycle:
0	The EMIF uses the original word address. The word read in this cycle is returned by the EMIF.	After the first access, the EMIF adds 1 to the previous word address to generate the second word address. The word read in this cycle is eventually discarded by the EMIF.
1	The EMIF uses the original word address minus one. The word read in this cycle is eventually discarded by the EMIF.	After the first access, the EMIF adds 1 to the previous word address to generate the second word address. The word read in this cycle is returned by the EMIF.

Figure 2−39. 16-Bit Data Write Accesses of 16-Bit-Wide External Memory

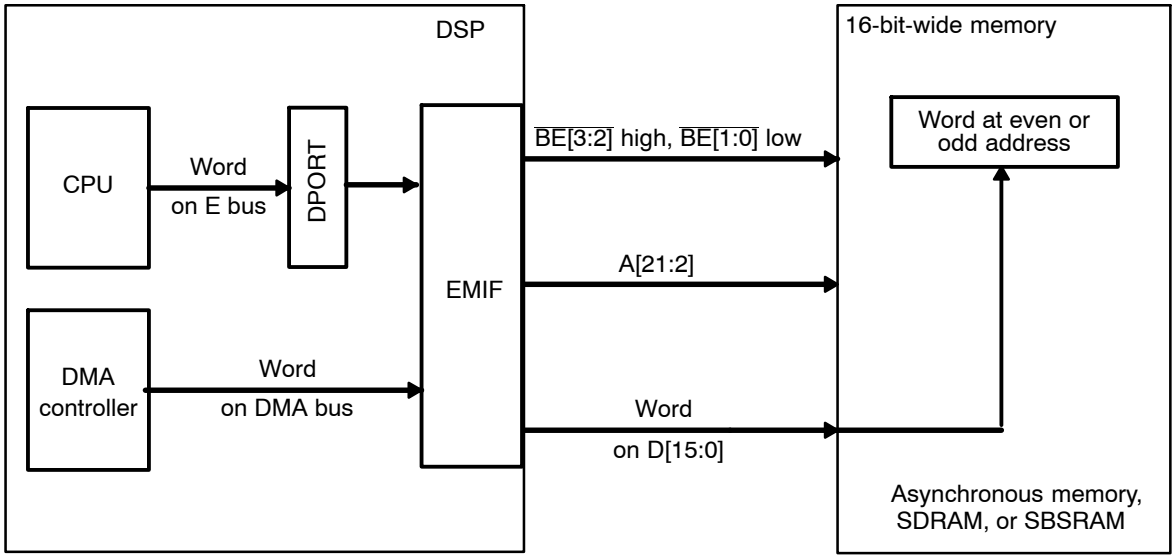
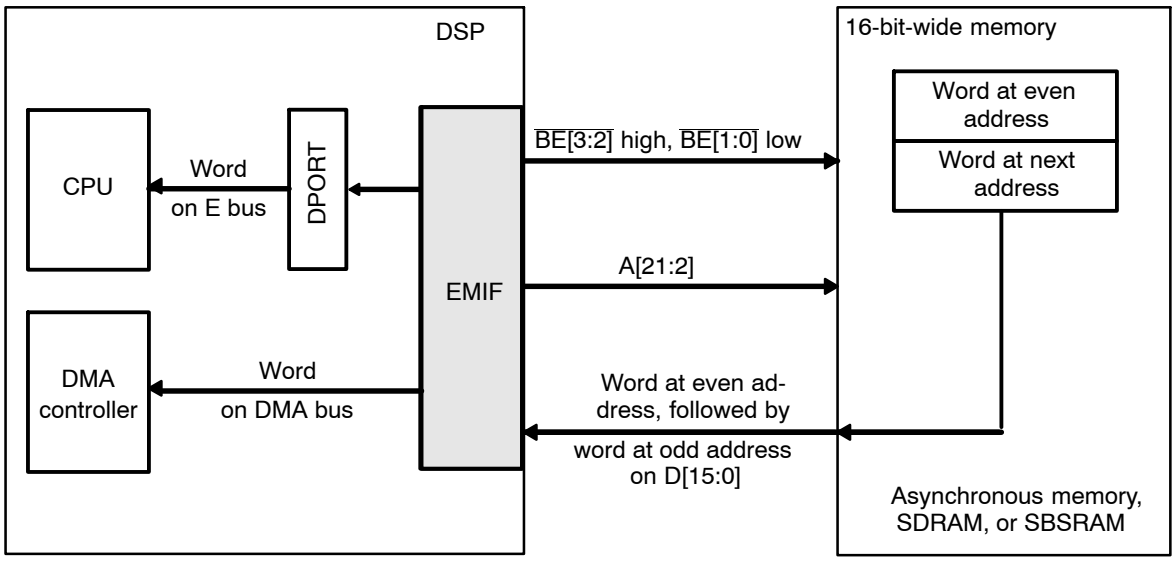


Figure 2−40. 16-Bit Data Read Accesses of 16-Bit-Wide External Memory



2.9.2.3 16-Bit Data Access of 8-Bit-Wide Memory

Figure 2−41 illustrates the data transfers involved in a 16-bit data write access of 8-bit-wide external memory. A 16-bit data read access from 8-bit-wide external memory is shown on Figure 2−42. The external address lines A[21:2] of the EMIF correspond to bits 19:0 of the internal data address. Data bus lines D[7:0] are used to transport the data between the DSP and the external memory. During an access, $\overline{BE}[3:1]$ stay high (inactive), and $\overline{BE0}$ is driven low.

When the EMIF makes 16-bit accesses in 8-bit-wide external memory, the number of byte accesses for read operations and write operations is different. When a word is written to external memory, the EMIF writes two bytes. However, when a word is read from external memory, the EMIF reads four 8-bit words from external memory (a total of 32-bits), and the desired word is generated in the DSP. The EMIF returns a word by combining two of the four bytes read from memory as described in Table 2−23.

Table 2−23. The Role of Internal Address Bit 1 During a 16-bit Data Read Access of 8-Bit-Wide External Memory

Internal Address Bit 1	Returned word is made up of :
0	The first and second bytes read from external memory.
1	The third and fourth bytes read from external memory.

Figure 2–41. 16-Bit Data Read Access of 8-Bit-Wide External Memory

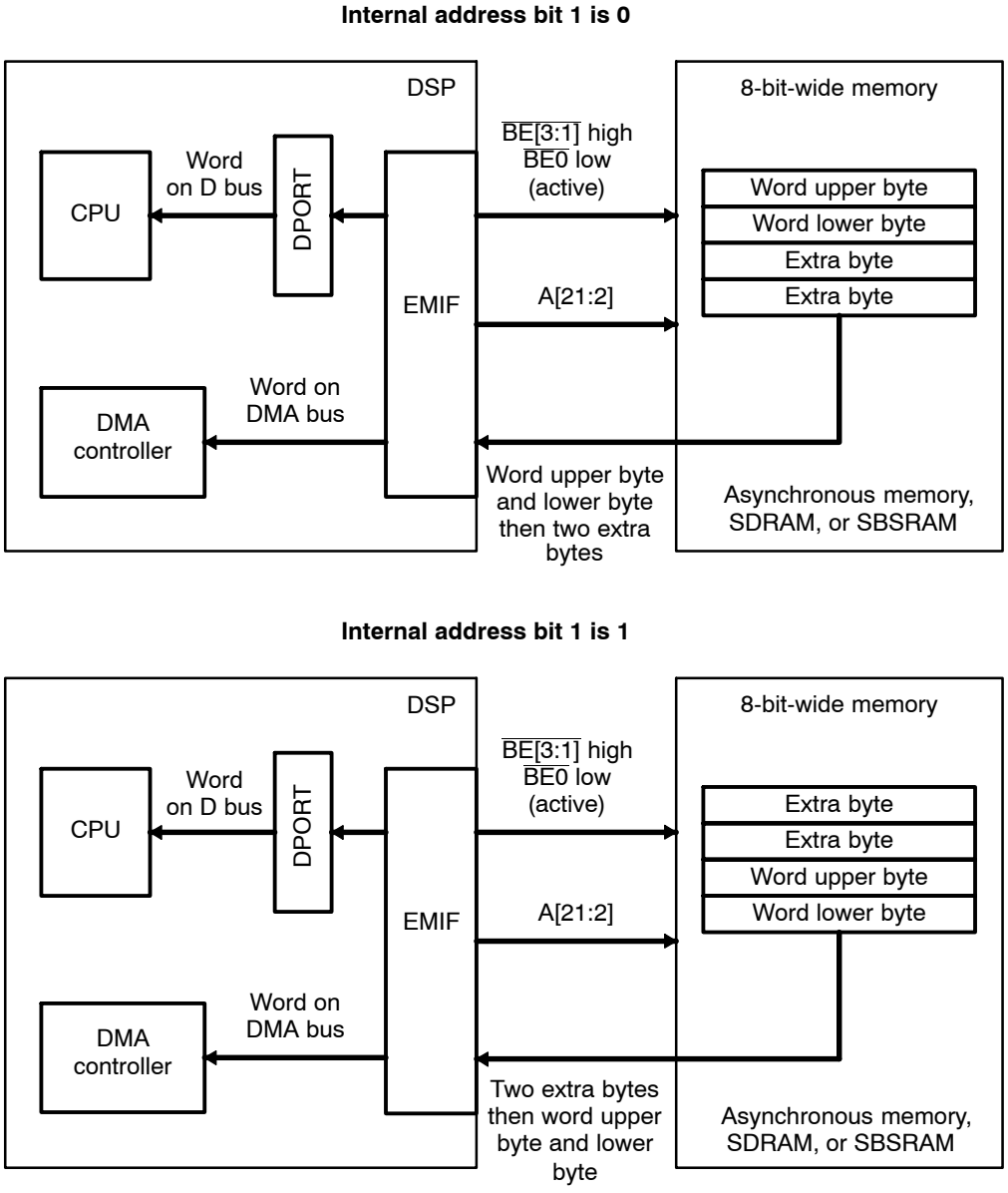
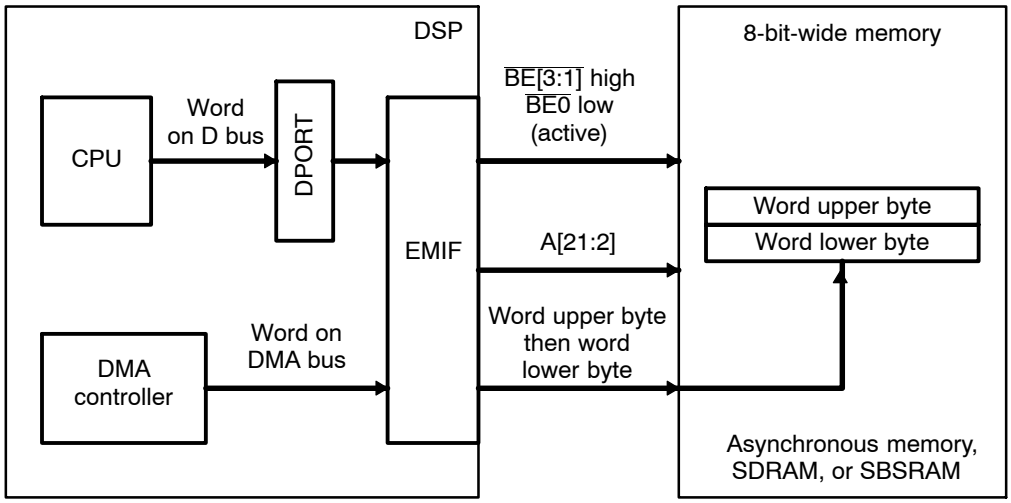


Figure 2−42. 16-Bit Data Write Access of 8-Bit-Wide External Memory



2.9.3 8-Bit Data Accesses

Some CPU instructions and DMA controller operations access 8-bit data (bytes). These byte accesses can be done in 8-bit-wide, 16-bit-wide, or 32-bit-wide memory. The CPU buses used are the D bus (for byte read operations) and the E bus (for byte write operations).

Note:

For 8-bit data read accesses, the EMIF always reads 32-bits from external memory and isolates the desired byte in the DSP. For 8-bit data write accesses, the EMIF modifies an individual byte.

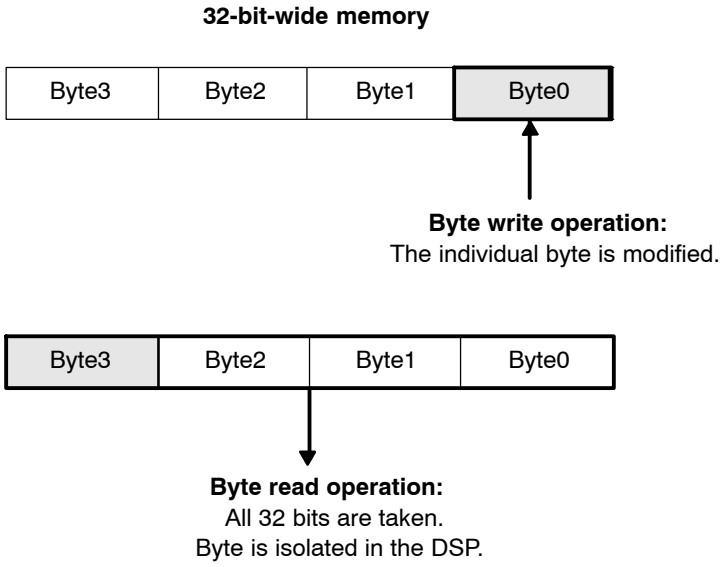
When doing an 8-bit data read, data is eventually discarded since the EMIF always reads 32-bits from external memory. This could significantly lower the overall system throughput since clock cycles are unnecessarily taken up to read unwanted data. However, when the DMA is used to read 8-bit data from external memory, data packing can be used to convert four adjacent 8-bit data read accesses into a single 32-bit data access. As described in section 2.9.1.3, no data is discarded when 32-bit data read accesses are done from 8-bit-wide external memory. More information on DMA data packing can be found in the *TMS320VC5501/5502 DSP Direct Memory Access (DMA) Controller Reference Guide (SPRU613)*.

A similar approach to the DMA can be used when using the CPU to read 8-bit data from external memory. Instead of doing consecutive 8-bit data accesses, the CPU could do 32-bit data accesses and parse the data internally. Depending on the EMIF settings, parsing could take a smaller number of clock cycles when compared to the number of clock cycles required to read three extra bytes from external memory.

2.9.3.1 8-Bit Data Access of 32-Bit-Wide Memory

As shown on Figure 2−43 below, when the EMIF makes 8-bit accesses in 32-bit-wide external memory, the actual width of the access is different for read operations and write operations. The EMIF reads the full width of the memory, and the desired byte is isolated in the DSP. For writes, the EMIF modifies an individual byte by activating only one byte enable signal.

Figure 2−43. Accessing 8-Bit Data in 32-Bit-Wide External Memory



External address lines A[21:2] correspond to bits 21-2 of the internal data address.

The way the EMIF writes 8-bit data to 32-bit-wide external memory is illustrated by Figure 2−44. Specifically, this figure shows the CPU or the DMA controller modifying the 8 LSBs (bits 7-0) of a memory location. 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 2−24).

The process for reading 8-bit data from 32-bit-wide external memory is illustrated by Figure 2−45. This figure shows four bytes being read from external memory. The EMIF uses bits 1 and 0 of the internal address to determine which byte is returned (see Table 2−25).

Table 2−24. 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

Table 2−25. The Role of Internal Address Bits 1-0 During an 8-Bit Data Read to 32-Bit-Wide External Memory

Internal Address Bits 1-0	Bits Returned by EMIF	Data Lines Used	Byte Enable Signal Levels
00	31-24 (the 8 MSBs)	D[31:0]	$\overline{BE[3:0]}$ low (active)
01	23-16	D[31:0]	$\overline{BE[3:0]}$ low (active)
10	15-8	D[31:0]	$\overline{BE[3:0]}$ low (active)
11	7-0 (the 8 LSBs)	D[31:0]	$\overline{BE[3:0]}$ low (active)

Figure 2−44. Writing to the 8 LSBs of a 32-Bit-Wide External Memory Location

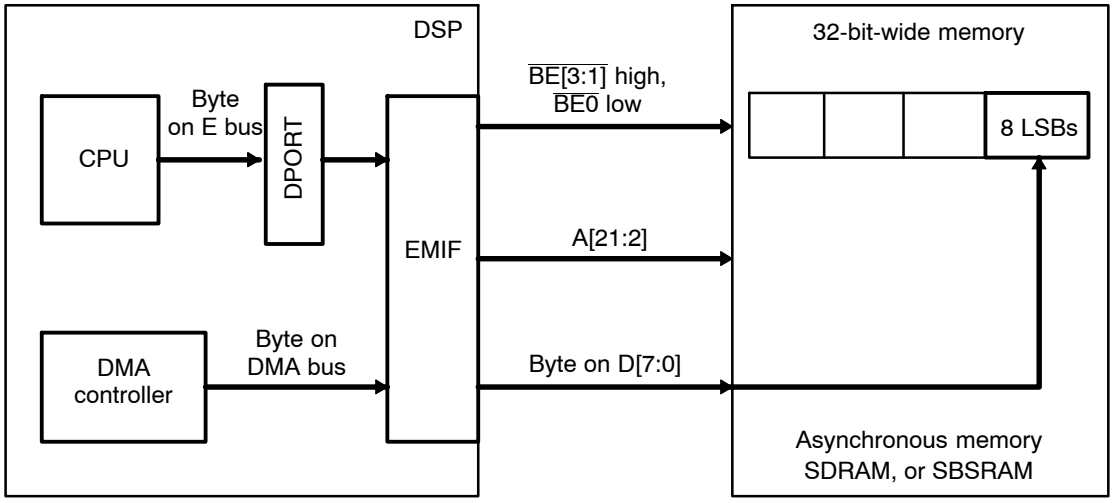
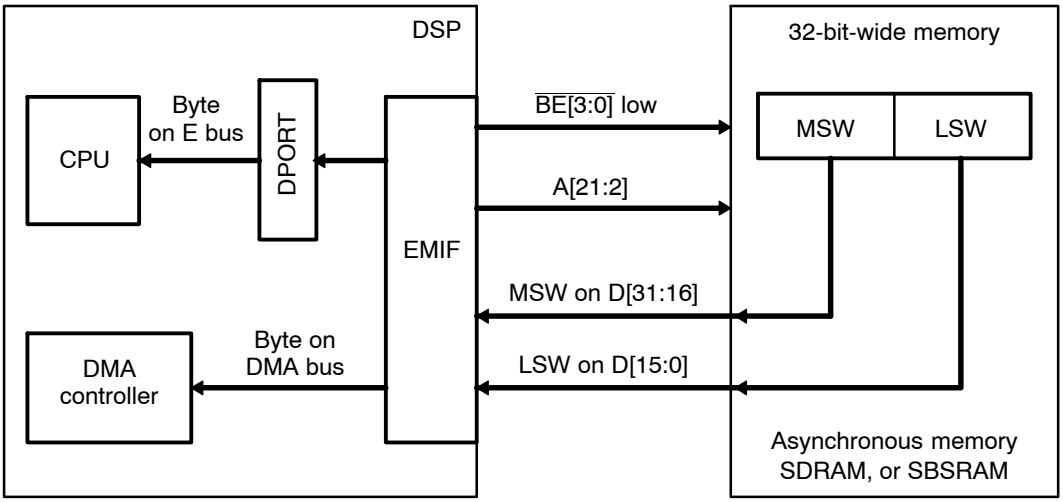


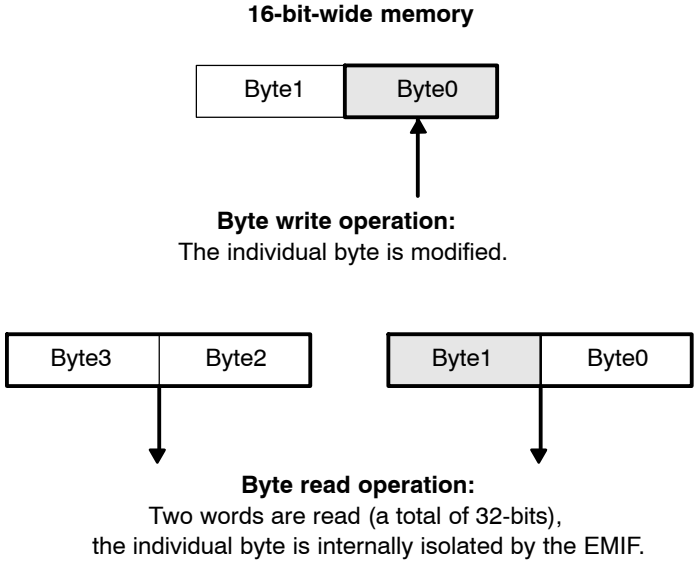
Figure 2−45. Reading 8 bits from a 32-Bit-Wide External Memory Location



2.9.3.2 8-Bit Data Accesses of 16-Bit-Wide Memory

As shown on Figure 2−46 below, when the EMIF makes 8-bit accesses in 16-bit-wide external memory, the actual number of accesses to external memory is different for read operations and write operations. The EMIF reads 8-bit data from 16-bit-wide memory the same way that it reads 32-bit data from 16-bit-wide memory (see 32-Bit Data Access of 16-Bit-Wide Memory on page 2-57). The EMIF reads two 16-bit values from external memory, and the desired byte is isolated in the DSP. For writes, the EMIF modifies an individual byte.

Figure 2−46. Accessing 8-Bit Data in 16-Bit-Wide External Memory



External address lines A[21:2] correspond to bits 20-1 of the internal data address.

Figure 2−47 illustrates the data transfers involved in an 8-bit write operation that uses 16-bit-wide external memory. Specifically, the figure shows the CPU or the DMA controller modifying the 8 MSBs (bits 15-8) of a memory location. 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 2−26).

For 8-bit reads from 16-bit-wide memory, the EMIF reads total of two 16-bit words (32 bits) from external memory. The first word read from external memory makes up the MSW of the 32-bit value, while the second word makes up the LSW. The EMIF isolates the desired byte from this 32-bit value based on bits 1 and 0 of the internal address as shown on Table 2−27. The rest of the data is discarded. The process for reading 8-bit data from 16-bit-wide external memory is illustrated by Figure 2−48.

Table 2−26. 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

Table 2−27. The Role of Internal Address Bits 1 and 0 During an 8-Bit Data Read from 16-Bit-Wide External Memory

Internal Address Bits 1-0	Bits Returned by EMIF	These bits correspond to:
00	31-24 (the 8 MSBs)	The upper byte of the first word read from external memory.
01	23-16	The lower byte of the first word read from external memory.
10	15-8	The upper byte of the second word read from external memory.
11	7-0 (the 8 LSBs)	The lower byte of the second word read from external memory.

Figure 2–47. 8-Bit Write Operation Using 16-Bit-Wide External Memory

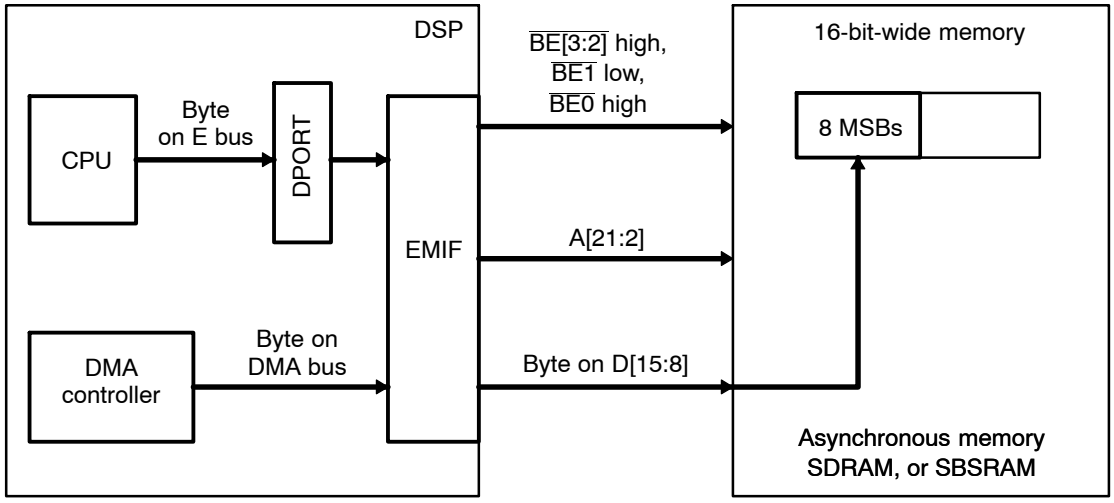
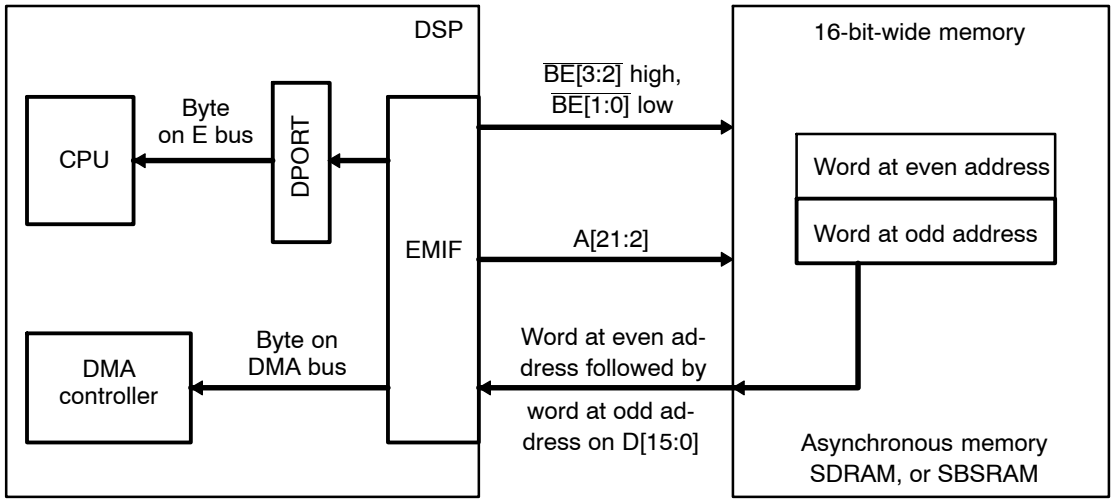


Figure 2–48. 8-Bit Read Operation Using 16-Bit-Wide External Memory



2.9.3.3 8-Bit Data Access of 8-Bit-Wide Memory

Figure 2−49 illustrates the data transfers involved in an 8-bit data write access of 8-bit-wide external memory. An 8-bit data read access from 8-bit-wide external memory is shown on Figure 2−50. The external address lines A[21:2] of the EMIF correspond to bits 19:0 of the internal data address. Data bus lines D[7:0] are used to transport the data between the DSP and the external memory. During an access, $\overline{BE[3:2]}$ stay high (inactive), and $\overline{BE0}$ is driven low.

When the EMIF makes 8-bit accesses in 8-bit-wide external memory, the number of byte accesses for read operations and write operations is different. When a byte is written to external memory, the EMIF writes an individual byte. However, when a byte is read from external memory, the EMIF reads four bytes from external memory (a total of 32-bits), and the desired byte is isolated in the DSP. The rest of the data is discarded. The byte that the EMIF returns is determined by bits 1 and 0 of the internal address as described in Table 2−28.

Table 2−28. The Role of Internal Address Bits 1 and 0 During an 8-Bit Data Read from 8-Bit-Wide External Memory

Internal Address Bits 1-0	Bits Returned by EMIF	These bits correspond to:
00	31-24 (the 8 MSBs)	The first byte read from external memory.
01	23-16	The second byte read from external memory.
10	15-8	The third byte read from external memory.
11	7-0 (the 8 LSBs)	The fourth byte read from external memory.

Figure 2–49. 8-Bit Write Operation Using 8-Bit-Wide External Memory

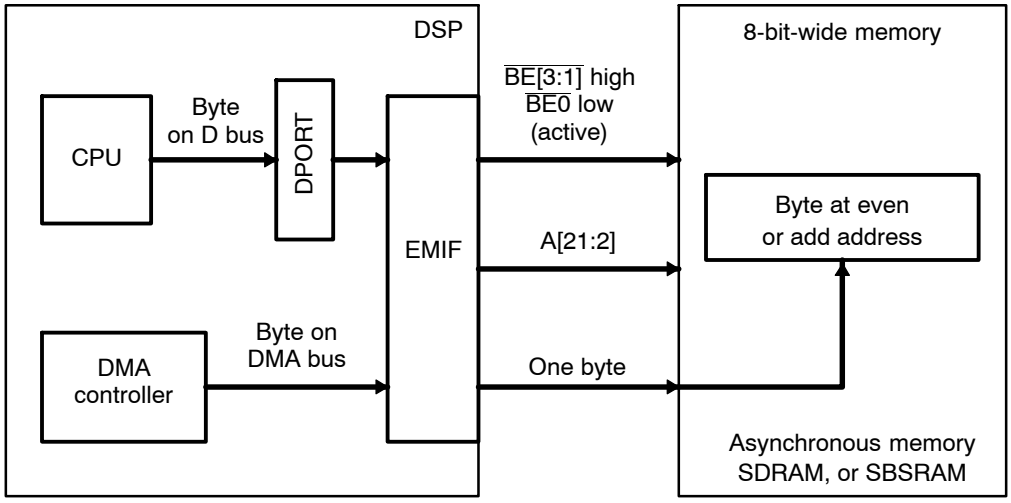
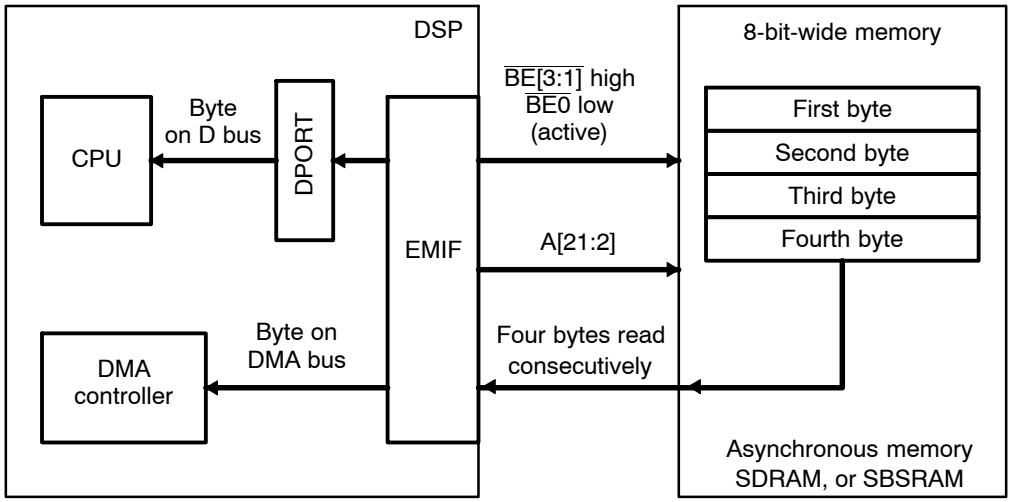


Figure 2–50. 8-Bit Read Operation Using 8-Bit-Wide External Memory



2.10 Write Posting

The EMIF has two write post registers, which can be freely associated with E and F bus writes. The write post registers are used to store the write address and data such that the CPU may be acknowledged in zero wait-state. The CPU is then free to carry on with the next access and the posted writes are run externally as slots become available. If the next access is not for the EMIF and is for an internal memory then that access is able to run concurrently with a slow external write.

As the write post registers can be freely associated (i.e., not dedicated to a particular internal bus) a patch of code which just comprises for example E bus writes, will benefit from two levels of write posting.

Write posting can be disabled via the WPE bit in the DPORT configuration register (DCR[8]). It may be useful during debug to disable write posting. When write posting has been disabled the internal write bus is acknowledged as the write is driven onto the external bus.

2.11 Memory Request Servicing

On the TMS230VC5501 and TMS320VC5502, the EMIF can receive memory requests from three different sources:

- ❏ DPORT
- ❏ IPORT
- ❏ DMA External Memory Port

The EMIF services requests from these three sources in a round-robin fashion.

The DPORT is used by the CPU's C, D, E, and F buses to access data in external memory. The DPORT prioritizes requests to external memory in the following manner.

Highest	E
	F
	D
Lowest	C

When executing code from external memory, the CPU uses the P bus to place requests to the instruction cache, which in turn accesses external memory using the IPORT.

2.12 EMIF Registers

Control of the external memory interface is maintained through a set of memory mapped registers within the EMIF. A write to any EMIF register is not complete until all pending EMIF accesses which use that register have completed. The memory-mapped registers are shown in Table 2−29. Note that it takes at least 6 cycles between the completion of a write to the register.

Table 2−29. EMIF Memory-Mapped Registers

Address	Name
0x0800	EMIF Global Control Register 1
0x0801	EMIF Global Control Register 2
0x0802	EMIF CE1 Space Control Register 1
0x0803	EMIF CE1 Space Control Register 2
0x0804	EMIF CE0 Space Control Register 1
0x0805	EMIF CE0 Space Control Register 2
0x0806− 0x0807	Reserved
0x0808	EMIF CE2 Space Control Register 1
0x0809	EMIF CE2 Space Control Register 2
0x080A	EMIF CE3 Space Control Register 1
0x080B	EMIF CE3 Space Control Register 2
0x080C	EMIF SDRAM Control Register 1
0x080D	EMIF SDRAM Control Register 2
0x080E	EMIF SDRAM Refresh Control Register 1
0x080F	EMIF SDRAM Refresh Control Register 2
0x0810	EMIF SDRAM Extension Register 1
0x0811	EMIF SDRAM Extension Register 2
0x0812− 0x0821	Reserved
0x0822	EMIF CE1 Secondary Control Register 1
0x0823	EMIF CE1 Secondary Control Register 2
0x0824	EMIF CE0 Secondary Control Register 1

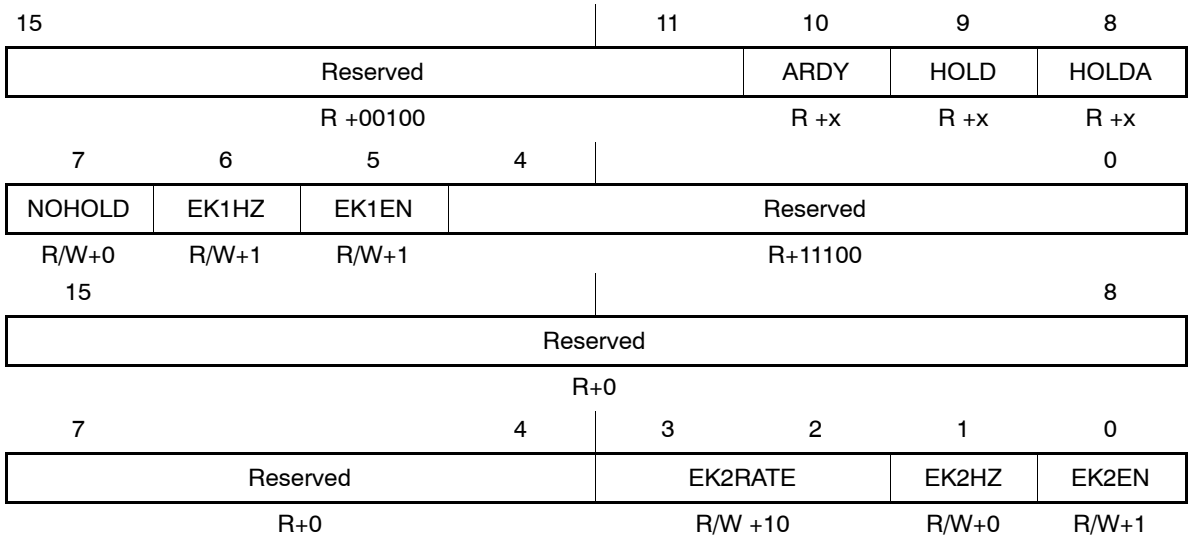
Table 2–29. EMIF Memory-Mapped Registers (Continued)

Address	Name
0x0825	EMIF CE0 Secondary Control Register 2
0x0826– 0x0827	Reserved
0x0828	EMIF CE2 Secondary Control Register 1
0x0829	EMIF CE2 Secondary Control Register 2
0x082A	EMIF CE3 Secondary Control Register 1
0x082B	EMIF CE3 Secondary Control Register 2
0x082C– 0x083F	Reserved
0x0840	EMIF CE Size Control Register 1
0x0841	EMIF CE Size Control Register 2

2.12.1 EMIF Global Control Registers

The EMIF Global Control Register () configures parameters common to all the CE spaces. It is recommended that this register should be set once during system initialization and not modified again.

Figure 2–51. EMIF Global Control Register 1 and 2 Diagram (0x0800-0x0801)



Legend: R/W-x = Read/Write-Reset value

Table 2–30. EMIF Global Control Register 1 Bit Field Descriptions

Bit	Field	Description
15-11	Reserved	
10	ARDY	ARDY =0: ARDY input is low. External device is not ready ARDY =1: ARDY input is high. External device is ready
9	HOLD	HOLD=0: $\overline{\text{HOLD}}$ input is low. External device is requesting EMIF HOLD=1: $\overline{\text{HOLD}}$ input is high. External device is not requesting EMIF
8	HOLDA	HOLDA=0: $\overline{\text{HOLDA}}$ output is low. External device owns EMIF HOLDA=1: $\overline{\text{HOLDA}}$ output is high. External device does not own EMIF
7	NOHOLD	External HOLD disable NOHOLD=0: hold enable NOHOLD=1: hold disable
6	EK1HZ	ECLKOUT1 High-Z control EK1HZ = 0, ECLKOUT1 is driven with value specified by EK*n*EN during Hold/Reset EK1HZ = 1, ECLKOUT1 is High-Z during Hold/Reset (must disable while clock is low) EK1HZ=1 (High-Z) by default.
5	EK1EN	ECLKOUT1 Enable (enabled by default) ECLKOUT1 is EMIF output clock at EMIF clock frequency. EMIF clock is either internal or external (user supplied) clock source. SDRAM interface synchronizes to ECLKOUT1. SBSRAM interface synchronizes to ECLKOUT1 or ECLKOUT2. 0 = disabled (held low) 1 = enabled
4-0	Reserved	

Table 2–31. EMIF Global Control Register 2 Bit Field Descriptions

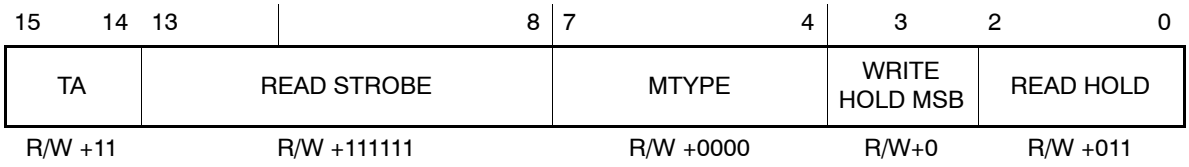
Bit	Field	Description
15-4	Reserved	
3-2	EK2RATE	ECLKOUT2 Rate (4x by default) 00 = 1x EMIF CLOCK period 01 = 2x EMIF CLOCK period 10 = 4x EMIF CLOCK period 11 = reserved ECLKOUT2 is only used by SBSRAM interface.
1	EK2HZ	ECLKOUT2 High-Z control EK2HZ = 0, ECLKOUT2 is driven with value specified by EK*n*EN during Hold/Reset EK2HZ = 1, ECLKOUT2 is High-Z during Hold/Reset (must disable while clock is low) EK2HZ is NOT High-Z by default.
0	EK2EN	ECLKOUT2 Enable (enabled by default) ECLKOUT2 is EMIF output clock at EMIF clock frequency div 1, 2, or 4. EMIF clock is either internal or external (user supplied) clock source. SBSRAM interface synchronizes to ECLKOUT1 or ECLKOUT2. 0 = disabled (held low) 1 = enabled

2.12.2 CE Space Control and Secondary Control Registers

The sixteen CE Space Control and Secondary Control Registers correspond to the four CE spaces supported by the EMIF. There are four CE space and secondary control registers per CE space. The *MType* bit field identifies the memory type for the corresponding CE space and contains a more detailed description of the configuration fields.

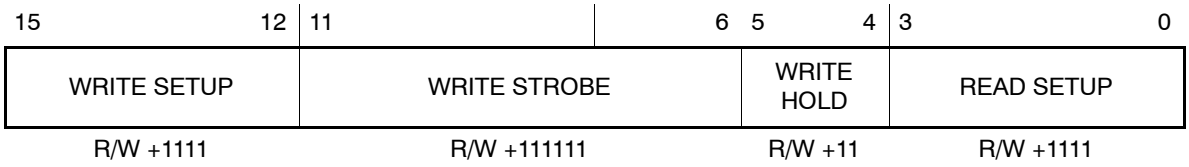
MTYPE cannot be changed on the fly. C5501/5502 does not support any two different types of memory in the same CE.

Figure 2–52. EMIF CE(0/1/2/3) Space Control Register 1 (0x0804, 0x0802, 0x0808, 0x080A)



Legend: R/W-x = Read/Write-Reset value

Figure 2–53. EMIF CE(0/1/2/3) Space Control Register 2 (0x0805, 0x0803, 0x0809, 0x080B)



Legend: R/W-x = Read/Write-Reset value

Table 2−32. EMIF CE(0/1/2/3) Space Control Register 1 Bit Field Descriptions

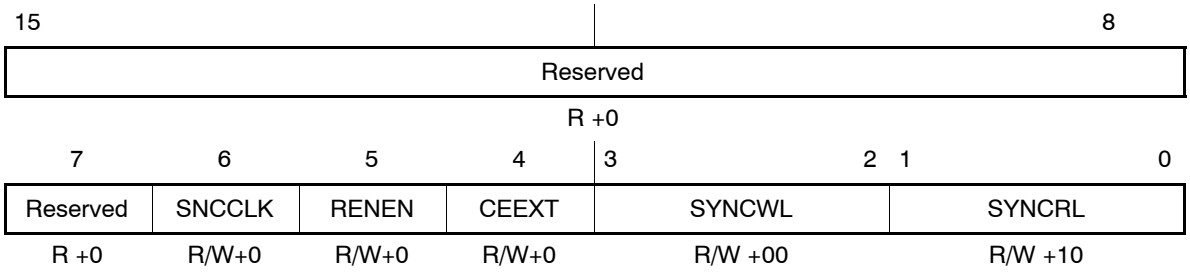
Bit	Field	Description
15-14	TA	Turn-Around time. Turn-Around time controls the number of ECLKOUT cycles between a read and a write, or between reads, to different CE spaces (asynchronous memory types only)
13-8	READ STROBE	Read strobe width. The width of read strobe ($\overline{ARE}$) in ECLKOUT1 clock cycles.
7-4	MTYPE	0000b: 8-bit-wide Asynchronous Interface 0001b: 16-bit-wide Asynchronous Interface 0010b: 32-bit-wide Asynchronous Interface 0011b: 32-bit-wide SDRAM 0100b: 32-bit-wide SBSRAM 0101b: Reserved 0110b: Reserved 0111b: Reserved 1000b: 8-bit-wide SDRAM 1001b: 16-bit-wide SDRAM 1010b: 8-bit-wide SBSRAM 1011b: 16-bit-wide SBSRAM 1100b: Reserved 1101b: Reserved 1110b: Reserved
3	WRITE HOLD MSB	Write hold width. See WRITE HOLD (Table 2−33).
2-0	READ HOLD	Read hold width. Number of ECLKOUT1 clock cycles that address (A[21-2]) and byte strobe enables ($\overline{BE}$[3-0]) are held after read strobe rises. For asynchronous, this is also the hold time of $\overline{AOE}$ after $\overline{ARE}$ rises.

Table 2−33. EMIF CE(0/1/2/3) Space Control Register2 Bit Field Descriptions

Bit	Field	Description
15-12	WRITE SETUP	Write setup width. Number ECLKOUT1 clock cycles of setup time for address(A[21-2]), chip enable ($\overline{CE}$[3:0]) and byte enables ($\overline{BE}$[3-0]) before write strobe falls.
11-6	WRITE STROBE	Write strobe width. The width of write strobe ($\overline{AWE}$) in ECLKOUT1 clock cycles.
5-4	WRITE HOLD	Write hold width. Number of ECLKOUT1 clock cycles that address (A[21-2]) and byte strobe enables ($\overline{BE}$[3-0]) are held after write strobe rises. Together, WRITE HOLD MSB and WRITE HOLD specify the Write Hold Width.
3-0	READ SETUP	Read setup width. Number ECLKOUT1 clock cycles of setup time for address(A[21-2]), chip enable ($\overline{CE}$[3:0]) and byte enables ($\overline{BE}$[3-0]) before read strobe falls. For asynchronous, this is also the setup time of $\overline{AOE}$ before $\overline{ARE}$ falls.

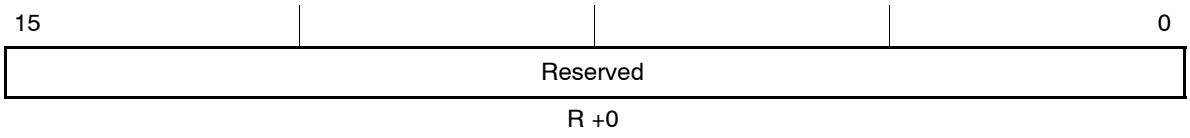
The secondary CE space control register is added for programmable synchronous interface. This register controls the cycle timing of programmable synchronous memory accesses, and the clock used for synchronization for the specific CE space. The CE space secondary control register (CExSEC) is shown in Figure 2−54 and Figure 2−55 and summarized in Table 2−34.

Figure 2−54. EMIF CE(0/1/2/3) Secondary Control Register 1 (0x0824, 0x0822, 0x0828, 0x082A)



Legend: R/W-x = Read/Write-Reset value

Figure 2−55. EMIF CE(0/1/2/3) Secondary Control Register 2 (0x0825, 0x0823, 0x0829, 0x082B)



Legend: R/W-x = Read/Write-Reset value

Table 2–34. EMIF CE(0/1/2/3) Secondary Control Register 1 Bit Field Descriptions

Bit	Field	Description
15-7	Reserved	
6	SNCCLK	SBSRAM I/F synchronized clock (Applies only to SBSRAM interface) SNCCLK = 0, control/data signals for this CE space are sync-ed to ECLKOUT1 SNCCLK = 1, control/data signals for this CE space are sync-ed to ECLKOUT2
5	RENEN	Read Enable Enable (Applies only to SBSRAM interface) 0 = ADS/REN signal acts as ADS signal 1 = ADS/REN signal acts as REN signal
4	CEEXT	CE extension register (Applies only to SBSRAM interface) CEEXT = 1, then on read cycles, the CE signal goes active when $\overline{SOE}$ goes active and stays active until $\overline{SOE}$ goes inactive. The $\overline{SOE}$ timing is controlled by SNCCRL.
3-2	SYNCWL	Synchronous interface cycle write latency (Applies only to SBSRAM interface) 00 = 0 cycle write latency 01 = 1 cycle write latency 10 = 2 cycle write latency 11 = 3 cycle write latency
1-0	SYNCRL	Synchronous interface cycle read latency (Applies only to SBSRAM interface) 00 = 0 cycle read latency 01 = 1 cycle read latency 10 = 2 cycle read latency 11 = 3 cycle read latency

Table 2–35. EMIF CE(0/1/2/3) Secondary Control Register 2 Bit Field Descriptions

Bit	Field	Description
15-0	Reserved	

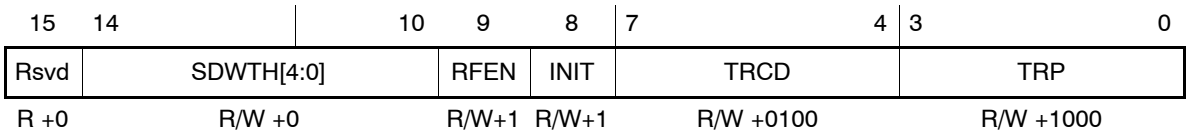
2.12.3 EMIF SDRAM Control Registers (SDCTL)

The SDRAM control registers control SDRAM parameters for all CE spaces, which specify an SDRAM memory type in the MTYPE field of its associated CE Space control register. Since the SDRAM control registers control all SDRAM spaces, each space must contain SDRAM with the same refresh, timing, and page characteristics. These registers should not be modified while accessing SDRAM.

Figure 2−56. EMIF SDRAM Control Register 1 (0x080C)

Legend: R/W-x = Read/Write-Reset value

Figure 2−57. EMIF SDRAM Control Register 2 (0x080D)



Legend: R/W-x = Read/Write-Reset value

Table 2−36. EMIF SDRAM Control Register 1 Bit Field Descriptions

Bit	Field	Description
15-12	TRC	Specifies ^{t}RC value of the SDRAM in EMIF clock cycles. TRC = (^{t}RC / ^{t}cyc) − 1 Valid values : [0,15] inclusive; 5, 6, 7, 8, 9, 10, 11
11-1	Reserved	
0	SLFRFR	Self-refresh mode. If SDRAM is used in the system: ❑ SLFRFR=0: Self-refresh mode disabled ❑ SLFRFR=1: Self-refresh mode enabled If SDRAM is not used in the system: ❑ SLFRFR=0: general purpose output SDCKE=1 ❑ SLFRFR=1: general purpose output SDCKE=0

Table 2–37. EMIF SDRAM Control Register 2 Bit Field Descriptions

Bit	Field	Description
15	Reserved	
14-10	SDWTH[4:0]	Determines the size of the SDRAM that are placed on the bus. The width of the data bus for a given CE space is determined by the MTYPE field in the CE control registers. SDWTH[4] 0 = 2 banks 1 = 4 banks SDWTH[3:2] 00 = 11 Row Address Pins 01 = 12 Row Address Pins 10 = 13 Row Address Pins 11 = Reserved SDWTH[1:0] 00 = 9 Column Address Pins 01 = 8 Column Address Pins 10 = 10 Column Address Pins 11 = Reserved
9	RFEN	Refresh enable RFEN=0: SDRAM refresh disabled RFEN=1: SDRAM refresh enabled
8	INIT	SDRAM initialization. Force initialization of all SDRAM present. INIT=0: No effect INTI=1: initialize SDRAM in each CE space configured for SDRAM. EMIF automatically changes INITback to 0 after SDRAM initialization performed.
7-4	TRCD	Specifies ^{t}RCD value of the SDRAM in EMIF clock cycles. TRCD = (^{t}RCD / ^{t}cyc) − 1 Valid values: [0:15] inclusive; 0, 1, 2, 3
3-0	TRP	Specifies ^{t}RP value of the SDRAM in EMIF clock cycles TRP = (^{t}RP / ^{t}cyc) − 1 Valid values: [0, 15] inclusive; 0, 1, 2, 3

SLFRFR bit forces the EMIF to place the external SDRAM in a low power mode, called Self Refresh, in which the SDRAM maintains valid data while consuming a minimal amount of power. This mode is entered when a 1 is written to the SLFRFR bit. When this bit is set, the RFEN bit must be cleared simultaneously by the user. As long as this bit field contains a 1, accesses must not be performed to SDRAM. It is the user's responsibility to ensure that the SLFRFR bit is turned on/off at appropriate times. To exit SLFRFR mode, a 0 must be written to the SLFRFR bit and then immediately read back.

If SDRAM is not in use by the system, then the SLFRFR bit can be used to control SDCKE as a general-purpose output. CKE =SLFRFR.

One additional consideration is that the EMIF automatically clears the INIT field to zero after it performs SDRAM initialization. When RESET goes inactive, none of the CE spaces are configured as SDRAM, so the INIT field quickly change from 1 to 0. The CPU should initialize all of the CE Space control registers and the SDRAM extension register before it sets the INIT bit back to 1.

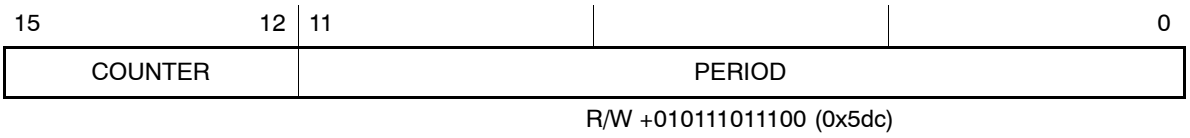
The SDWTH field defines the number of banks in the SDRAM, the size of the row address, and the size of the column address.

SDWTH does not determine the width of the data bus for a given CE space. Rather, it determines the size of the SDRAMs that are placed on the bus. The width of the data bus for a given CE space is determined by the MTYPE field in the CE control registers.

2.12.4 EMIF SDRAM Refresh Control Register (SDRFR)

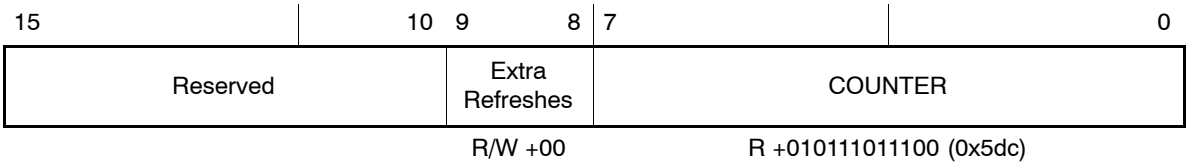
The SDRAM refresh period register controls the refresh period in terms of EMIF clock cycles. Note that C5501/5502 EMIF supports self refresh.

Figure 2−58. EMIF SDRAM Refresh Control Register 1 (0x080E)



Legend: R/W-x = Read/Write-Reset value

Figure 2−59. EMIF SDRAM Refresh Control Register 2 (0x080F)



Legend: R/W-x = Read/Write-Reset value

Table 2−38. EMIF SDRAM Refresh Control Register 1 Bit Field Descriptions

Bit	Field	Description
15-12	COUNTER	Current counter value (read-only)
11-0	PERIOD	Refresh counter preset value in ECLKOUT1 cycles

Table 2−39. EMIF SDRAM Refresh Control Register 2 Bit Field Descriptions

Bit	Field	Description
15-10	Reserved	
9-8	Extra Refreshes	The number of refreshes that take place when the counter reaches zero. “00” for 1, “01” for 2, “10” for 3, and “11” for 4.
7-0	COUNTER	Current counter value (read-only)

The extra refresh field controls the number of refreshes that take place when the counter reaches zero (“00” for 1, “01” for 2, “10” for 3, and “11” for 4). As an example, since all banks must be deactivated to perform a refresh, it might be desirable to perform two refreshes half as often.

All refresh requests are considered high priority. When it is time to refresh, the refresh is performed immediately (though transfers in progress are allowed to complete). All banks must be deactivated before a refresh command is issued. When the refresh command is complete, the banks are not restored to their state before refresh.

2.12.5 EMIF SDRAM Extension Register (SDEXT)

Any write in this register brings about SDRAM initialization in each CE space configured for SDRAM.

Figure 2−60. EMIF SDRAM Extension Register 1 (0x0810)

15	14	12	11	10	9	8
R2WDQM	RD2WR		RD2DEAC		RD2RD	THZP
0	R/W +101		R/W +11		R/W+1	R/W +10

7	6	5	4	3	1	0
THZP	TWR		TRRD	TRAS		TCL
R/W +10	R/W +01		R/W+1	R/W +111		R/W+1

Legend: R/W-x = Read/Write-Reset value

Figure 2−61. EMIF SDRAM Extension Register 2 (0x0811)

15						
Reserved						
R +0						

5	4	3	2	1	0
Reserved	WR2RD	WR2DEAC		WR2WR	R2WDQM
R +0	R/W+1	R/W +01		R/W+1	R/W+1

Legend: R/W-x = Read/Write-Reset value

Table 2–40. EMIF SDRAM Extension Register 1 Bit Field Descriptions

Bit	Field	Description
15	R2WDQM	Specifies the number of ECLKOUT1 cycles that BEx outputs must be high before the data driven for a write command interrupts a read command. The R2WDQM is a 2-bit field contained in both SDEXT1 and SDEXT2. Bit 15 of SDEXT1 is the least-significant bit and bit 0 of SDEXT2 is the most-significant bit. R2WDQM = (# of cycles BEx high) – 1 Valid values : 1, 2, 3
14-12	RD2WR	Specifies the number of ECLKOUT1 cycles between a READ command (CAS) and the first data driven for a write command. Note that the WRITE CAS occurs one cycle later. RD2WR = (# of cycles READ to WRITE) – 1 Valid values : 3, 4, 5
11-10	RD2DEAC	Specifies number of cycles between READ to DEAC/DCAB of the SDRAM in ECLKOUT1 cycles. RD2DEAC = (# of cycles READ to DEAC/DCAB) – 1 Valid values: 1,2,3
9	RD2RD	Specifies number of cycles between READ to READ command (same CE space) of the SDRAM in ECLKOUT1 cycles. RD2RD = 0, READ to READ = 1 ECLKOUT1 Cycle RD2RD = 1, READ to READ = 2 ECLKOUT1 Cycle
8-7	THZP	Specifies ^{t}HZP value of the SDRAM in ECLKOUT1 cycles. THZP = ^{t}HZP – 1 Valid values : 0, 1, 2, 3
6-5	TWR	Specifies ^{t}WR value of the SDRAM in ECLKOUT1 cycles. TWR = ^{t}WR – 1 Valid values : 0, 1, 2, 3
4	TRRD	Specifies ^{t}RRD value of the SDRAM in ECLKOUT1 cycles. TRRD = 0, then ^{t}RRD = 2 ECLKOUT1 cycles TRRD = 1, then ^{t}RRD = 3 ECLKOUT1 cycles

Table 2–40. EMIF SDRAM Extension Register 1 Bit Field Descriptions (Continued)

Bit	Field	Description
3-1	TRAS	Specifies ^{t}RAS value of the SDRAM in ECLKOUT1 cycles. TRAS = ^{t}RAS – 1 Valid values: 0, 1, 2, 3, 4, 5, 6, 7
0	TCL	Specified CAS latency of the SDRAM in ECLKOUT1 cycles. TCL = 0, CAS latency = 2 ECLKOUT1 cycles TCL = 1, CAS latency = 3 ECLKOUT1 cycles

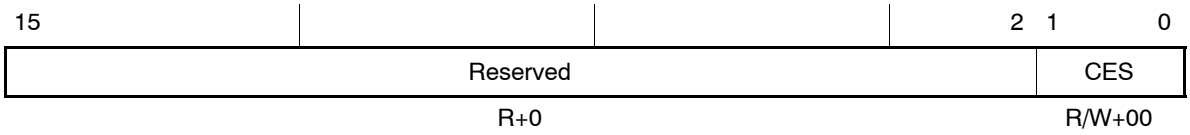
Table 2−41. EMIF SDRAM Extension Register 2 Bit Field Descriptions

Bit	Field	Description
15-5	Reserved	
4	WR2RD	Specifies minimum number of cycles between WRITE to READ command of the SDRAM in ECLKOUT1 cycles WR2RD = (# of cycles WRITE to READ) − 1 Valid values : 0, 1
3-2	WR2DEAC	Specifies minimum number of cycles between WRITE to DEAC/DCAB command of the SDRAM in ECLKOUT1 cycles. WR2DEAC = (# of cycles WRITE to DEAC/DCAB) − 1 Valid values : 1
1	WR2WR	Specifies minimum number of cycles between WRITE to WRITE command of the SDRAM in ECLKOUT1 cycles. WR2WR = (# of cycles WRITE to WRITE) − 1 Valid values : 0, 1
0	R2WDQM	Specifies the number of ECLKOUT1 cycles that BEx outputs must be high before the data driven for a write command interrupts a read command. The R2WDQM is a 2-bit field contained in both SDEXT1 and SDEXT2. Bit 15 of SDEXT1 is the least-significant bit and bit 0 of SDEXT2 is the most−significant bit. R2WDQM = (# of cycles BEx high) − 1 Valid values : 1, 2, 3

2.12.6 EMIF CE Size Control Register (CESCR)

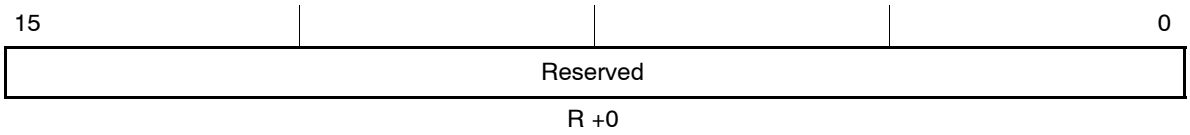
Any write in this register selects either 4Mbyte per each CE space (CE(0-3)), 8Mbyte per each CE space (CE(0,2)), or 16Mbyte per each CE space (CE0).

Figure 2−62. EMIF CE Size Control Register 1 (0x0840)



Legend: R/W-x = Read/Write-Reset value

Figure 2−63. EMIF CE Size Control Register 2 (0x0841)



Legend: R/W-x = Read/Write-Reset value

Table 2−42. EMIF CE Size Control Register 1 Bit Field Descriptions

Bit	Field	Description	
15-2	Reserved		
1-0	CES	CE size.	
		CES=00	CE0: 0x000000-0x3FFFFF CE1: 0x400000-0x7FFFFF CE2: 0x800000-0xBFFFFF CE3: 0xC00000-0xFFFFFF
		CES=01	CE0: 0x000000-0x7FFFFF CE1: not available CE2: 0x800000-0xFFFFFF CE3: not available
		CES=10	CE0: 0x000000-0xFFFFFF CE1: not available CE2: not available CE3: not available
		CES=10	Reserved

Table 2−43. EMIF CE Size Control Register2 Bit Field Descriptions

Bit	Field	Description
15-0	Reserved	

This page is intentionally left blank.

Appendix A

Revision History

Page	Additions/Modifications/Deletions
2-92	Changed R2WDQM bit description in Table 2–40.
2-94	Changed R2WDQM bit description in Table 2–41.
2-94	Changed second occurrence of Table 2–40 to Table 2–41.

Index

A

C

D

E

H

I

M

N

P

R

S

T

W

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