TMS320C6000 DSP
Enhanced Direct Memory Access (EDMA) Controller
Reference Guide

Literature Number: SPRU234C
November 2006

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
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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

Copyright © 2006, Texas Instruments Incorporated

Preface

Read This First

About This Manual

This document describes the operation of the enhanced direct memory access (EDMA) controller in the digital signal processors (DSPs) of the TMS320C6000™ DSP family. This document also describes the quick DMA (QDMA) used for fast data requests by the CPU. For operation and registers unique to the TMS320C621x/C671x EDMA, see Chapter 3. For operation and registers unique to the TMS320C64x™ EDMA, see Chapter 4.

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h. For example, the following number is 40 hexadecimal (decimal 64): 40h.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure designate a bit that is used for future device expansion.

Related Documentation From Texas Instruments

The following documents describe the C6000™ 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.

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

TMS320C6000 CPU and Instruction Set Reference Guide (literature number SPRU189) describes the TMS320C6000™ CPU architecture, instruction set, pipeline, and interrupts for these digital signal processors.

TMS320C6000 DSP Peripherals Overview Reference Guide (literature number SPRU190) describes the peripherals available on the TMS320C6000™ DSPs.

TMS320C6000 Technical Brief (literature number SPRU197) gives an introduction to the TMS320C62x™ and TMS320C67x™ DSPs, development tools, and third-party support.

TMS320C64x Technical Overview (SPRU395) gives an introduction to the TMS320C64x™ DSP and discusses the application areas that are enhanced by the TMS320C64x VelociTI™.

TMS320C6000 Programmer's Guide (literature number SPRU198) describes ways to optimize C and assembly code for the TMS320C6000™ DSPs and includes application program examples.

TMS320C6000 Code Composer Studio Tutorial (literature number SPRU301) introduces the Code Composer Studio™ integrated development environment and software tools.

Code Composer Studio Application Programming Interface Reference Guide (literature number SPRU321) describes the Code Composer Studio™ application programming interface (API), which allows you to program custom plug-ins for Code Composer.

TMS320C6x Peripheral Support Library Programmer's Reference (literature number SPRU273) describes the contents of the TMS320C6000™ peripheral support library of functions and macros. It lists functions and macros both by header file and alphabetically, provides a complete description of each, and gives code examples to show how they are used.

TMS320C6000 Chip Support Library API Reference Guide (literature number SPRU401) describes a set of application programming interfaces (APIs) used to configure and control the on-chip peripherals.

Trademarks

Code Composer Studio, C6000, C62x, C64x, C67x, TMS320C6000, TMS320C62x, TMS320C64x, TMS320C67x, and VelociTI are trademarks of Texas Instruments.

Contents

Figures

Tables

Chapter 1

Overview

This chapter provides an overview and describes the common operation of the enhanced direct memory access channel controller (EDMACC) in the digital signal processors (DSPs) of the TMS320C6000™ DSP family. This chapter also describes the quick DMA (QDMA) used for direct data requests by the CPU. For an overview and description of the enhanced direct memory access transfer controller (EDMATC), see Chapter 2.

For operation and registers unique to the TMS320C621x/C671x EDMA, see Chapter 3. For operation and registers unique to the TMS320C64x™ EDMA, see Chapter 4.

Topic **Page**

1.1 Overview

The enhanced direct memory access (EDMA) controller handles all data transfers between the level-two (L2) cache/memory controller and the device peripherals on the C621x/C671x/C64x DSP, as shown in Figure 1−1. These data transfers include cache servicing, non-cacheable memory accesses, user-programmed data transfers, and host accesses.

The EDMA controller in the C621x/C671x/C64x DSP has a different architecture from the previous DMA controller in the C620x/C670x devices. The EDMA includes several enhancements to the DMA, such as 64 channels for the C64x DSP or 16 channels for the C621x/C671x DSP, with programmable priority, and the ability to link and chain data transfers. The EDMA allows movement of data to/from any addressable memory spaces, including internal memory (L2 SRAM), peripherals, and external memory.

The enhanced direct memory access consists of two primary components:

1) Transfer controller (EDMATC) handles all data transfers between the level-two (L2) cache/memory controller and the device peripherals on the C621x/C671x/C64x DSP, as shown in Figure 1−1 These data transfers include EDMA channel controller transfers, cache accesses to/from EMIF range, noncacheable memory accesses, and master peripheral accesses.

2) Channel controller (EDMACC) is the user-programmable portion of the EDMA that supports a flexible and powerful set of transfers, including 1D and 2D transfers; flexible triggering including event triggered transfers, chained transfers, and CPU triggered transfers; and flexible reload and addressing modes that support ping-pong buffers, circular buffering, frame extraction, and sorting.

Figure 1−1. TMS320C621x/C671x/C64x DSP Block Diagram

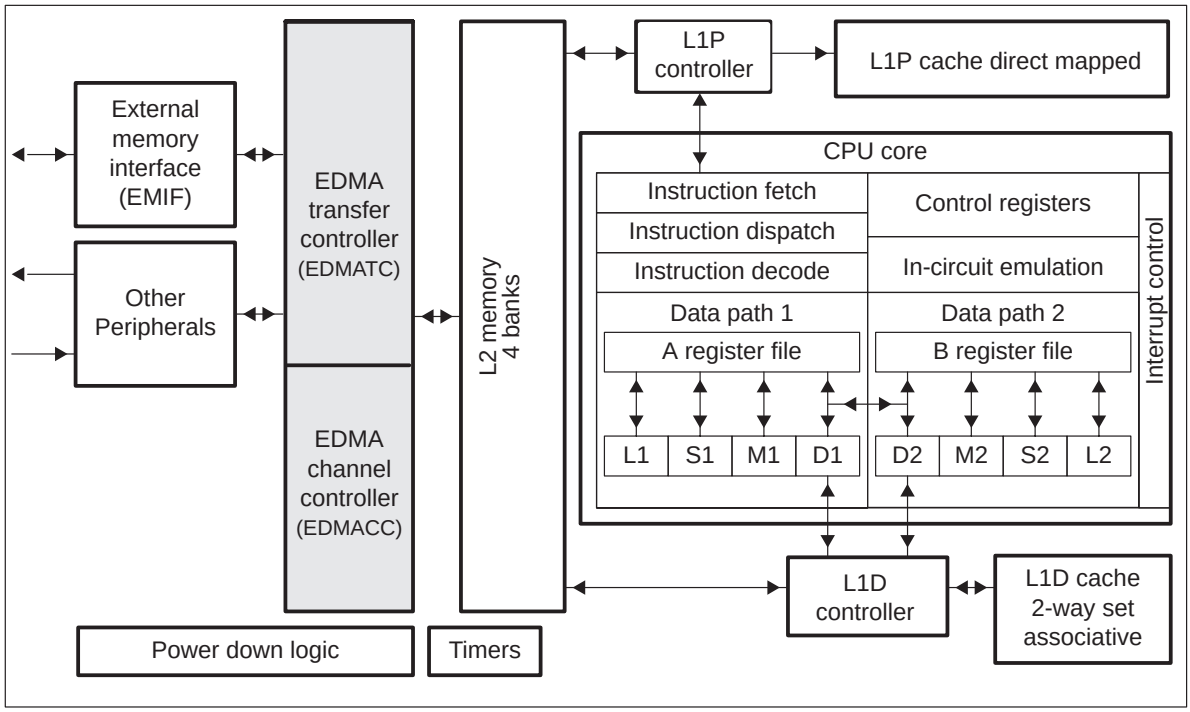


Figure 1−2 shows a block diagram of the EDMA channel controller. The EDMA channel controller (EDMACC) comprises:

- Parameter RAM (PaRAM): Maintains parameter entries for channel and reload parameter sets. You write to the PaRAM to setup transfer context for desired channels and link parameter sets. EDMACC processes entries based on a trigger event and submits transfer request (TR) to the EDMA transfer controller (EDMATC).
- Event and interrupt processing registers: Allows you to enable/disable events; enable trigger types; enable/disable interrupt conditions; and clear and process interrupts based on completion of DMAs.
- Completion detect: Completion detect block detects completion of transfers within EDMATC. Completion of transfers can optionally be used to cause submission of new transfers (chaining) or generate interrupts to the CPU via EDMA_INT.

Table 1−1 summarizes the difference between the C6000 EDMAs.

Figure 1−2. EDMA Channel Controller Block Diagram

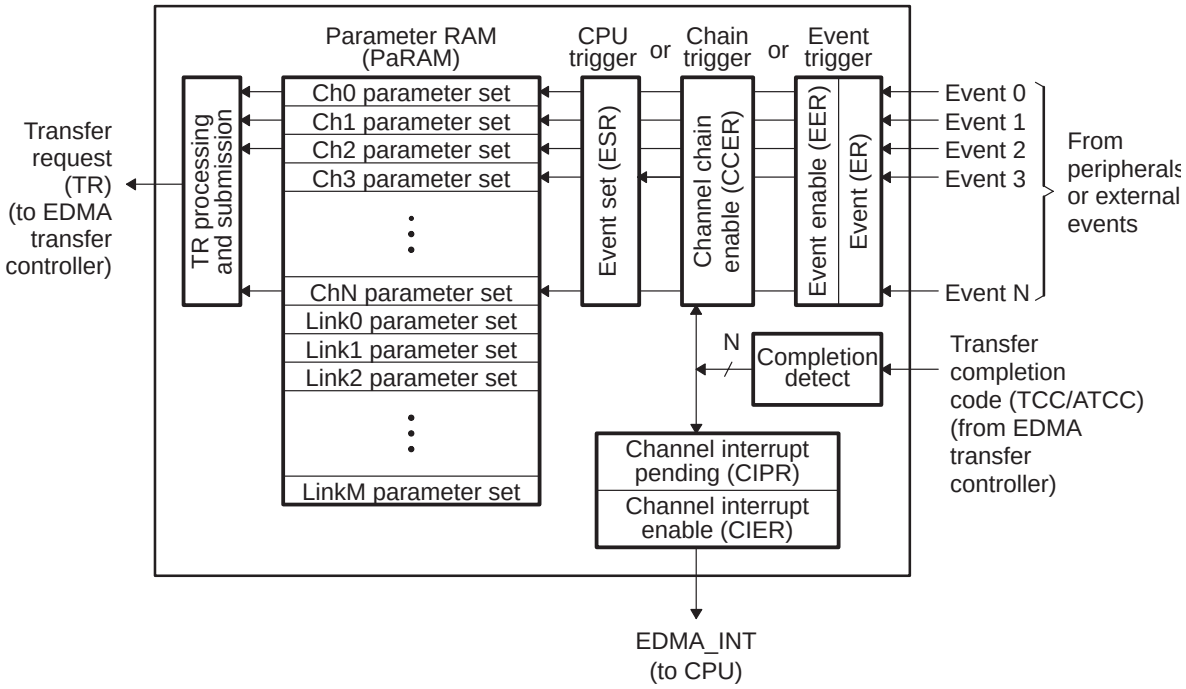


Table 1−1. Differences Between the C621x/C671x and C64x EDMA

Features	C621x/C671x EDMA	C64x EDMA
Alternate transfer complete chaining and interrupt	Does not apply.	Supported
CIPR, CIER, CCER, ER, EER, ECR, ESR	Each of these registers supports 16 channels.	Each of these registers supports 64 channels and is expanded into two registers (low and high).
EDMA clock rate	EDMA clock rate equals CPU clock rate.	EDMA clock rate equals one half of CPU clock rate (CPU/2).
EDMA transfers possible on all priority queues	EDMA cannot transfer on Q0.	EDMA transfers possible on all priority queues.
Event polarity selection	Does not apply.	Supported.
L2 controller transfers possible on all priority queues	L2 controller transfers on Q0 only.	L2 controller transfers possible on all priority queues.
Number of channels	16 channels.	64 channels.
Programmable priority queue allocation (PQAR0−3)	Does not apply.	Supported.
Programmable event polarity (EPRL, EPRH)	Does not apply.	Supported.
Supports peripheral device transfers	Does not apply.	Supported.
Transfer chaining on channels	Only channels 8 to 11 can be chained.	All channels can be chained.

1.2 EDMA Terminology

The following definitions help in understanding some of the terms used in this document:

- **Enhanced Direct Memory Access (EDMA) Controller:** Consists of the EDMA transfer controller (EDMATC) and the EDMA channel controller (EDMACC).

- **EDMA Transfer Controller (EDMATC):** The central data transfer engine of the EDMA. The EDMATC concurrently performs data transfers for the first queued transfer request (TR) of each transfer controller priority queue. The number of TR queues (and thus concurrent transfers) is device dependent.

- **EDMA Channel Controller (EDMACC):** The user-programmable portion of the EDMA. The EDMACC consists of a parameter RAM (PaRAM) (and associated enable/disable registers) that can be programmed with many active (channel parameter sets) and reload parameter sets that can be initiated via external events, peripheral events, or CPU synchronization. The number of channel/link parameter sets is device dependent.

- **Transfer Request (TR):** A request for data movement that is submitted by all transfer controller requestors to the transfer controller. A TR includes source address, destination address, element count, options, etc. For the EDMA channel controller, a TR is submitted based on a synchronization event.

- **Quick DMA (QDMA):** A programmable DMA parameter set that is local to the CPU, rather than located in the EDMA channel controller. QDMA allows direct (and quick) CPU-initiated DMA submission without having to submit requests to the EDMA channel controller.

- **Element-Synchronized Transfer:** An element-synchronized transfer submits a transfer request (TR) for a single element when a synchronization event is received. The EDMA channel controller updates the source address, destination address, element count, and frame count after each TR.

- **Frame-Synchronized Transfer:** A frame-synchronized transfer submits a transfer request (TR) for a single frame of element count elements when a synchronization event is received. The EDMA channel controller updates the source address, destination address, and frame count after each TR. Elements and frames are spaced by a programmable index.

- **Array-Synchronized Transfer:** An array-synchronized transfer submits a transfer request (TR) for a single array of element count elements when a synchronization event is received. The EDMA channel controller updates the source address, destination address, and array count after each TR. Elements are always contiguous and frames are spaced by a programmable index.
- **Block-Synchronized Transfer:** A block-synchronized transfer submits a transfer request (TR) for an array count arrays of element count elements when a synchronization event is received. The EDMA channel controller does not perform any address or count updates. Elements are always contiguous and frames are spaced by a programmable index.
- **Parameter RAM (PaRAM):** User-programmable RAM that stores channel parameter sets and link parameter sets.
- **Parameter Set:** A 24-byte long EDMA channel transfer definition. Each parameter set consists of six 4-byte parameter set entries.
- **Parameter Set Entry:** One of the 4-byte components of a parameter set. This includes source address, destination address, options, count, index, and reload entries.
- **Channel Parameter Set:** One of the 16 (621x/671x DSP) or 64 (64x DSP) parameter sets that can be initiated via a synchronization event. Note that unused channel parameter sets can also be used for linking.
- **Link Parameter Set:** One of the parameter sets that are strictly used for linking and are not one of the channel parameter sets.
- **Trigger Event:** Action that causes the EDMA channel controller to submit a transfer request (TR). Synchronization events include CPU-trigger, event-trigger, and chain-trigger.
- **Element:** The smallest unit of a DMA transfer. An element can be 8-bit, 16-bit, or 32-bit.
- **Frame:** A group of elements comprise a frame. A frame can have staggered or contiguous elements. Frame is used in context with 1-dimensional (1D) transfer.
- **Array:** A group of contiguous elements comprise an array. The elements in an array cannot be spaced by an element index. Array is used in context with 2-dimensional (2D) transfer.
- **Block:** A group of arrays or frames form a block. For 1-dimensional (1D) transfers, a group of frames form a block. For 2-dimensional (2D) transfers, a group of arrays form a block.

- **1-dimensional (1D) transfer:** A group of frames comprise a 1D block. The number of frames (FRMCNT) in a block can range from 1 to 65536. The number of elements (ELECNT) per frame can range from 1 to 65535. Either elements or full frames can be transferred at a time.
- **2-dimensional (2D) transfer:** A group of arrays comprise a 2D block. The first dimension is the number of contiguous elements in an array, and the second dimension is the number of such arrays. The number of arrays (FRMCNT) in a block can range from 1 to 65536. Either arrays or the entire block can be transferred at a time.

1.3 Parameter RAM Overview

Unlike the C620x/C670x DMA controller, which is a register-based architecture, the EDMA channel controller (EDMACC) is a RAM-based architecture. The parameter RAM (PaRAM) consists of a device-dependent number of parameter sets, ranging from 85 parameter sets (2-KByte RAM) to 213 parameter sets (5-KByte RAM).

Parameter sets are further defined as channel parameter sets or link parameter sets. The C621x/C671x EDMA supports 16 channels; thus providing 16 channel parameter sets. The C64x EDMA supports 64 channels; thus providing 64 channel parameter sets. The remaining PaRAM sets are link parameter sets.

Channel parameter sets can be triggered by various methods, including event trigger, chain trigger, or CPU trigger. Channel parameter sets that are not used as an active channel can also be used as a link entry. This is detailed in section 1.11. Link parameter sets cannot be directly submitted via a trigger. Instead, the link parameter sets can be automatically copied (or linked) onto one of the channel parameter sets upon completion of the parameter set for the given channel. This allows the CPU to setup and manage ping-pong buffers, circular buffers, etc, while a transfer is ongoing by updating the link parameter sets as necessary.

Table 1−2 shows the PaRAM format for a C6416 DSP, which consists of 64 channels and 85 total parameter RAM entries (64 channel parameter sets and 21 link parameter sets). Each parameter entry of an EDMA event is organized into six 32-bit words or 24 bytes, as shown in Figure 1−3 and described in Table 1−3.

Table 1−2. EDMA Parameter RAM Contents—C6416 DSP

Address	Parameters
01A0 0000h to 01A0 0017h	Parameters for event 0 (6 words)
01A0 0018h to 01A0 002Fh	Parameters for event 1 (6 words)
01A0 0030h to 01A0 0047h	Parameters for event 2 (6 words)
01A0 0048h to 01A0 005Fh	Parameters for event 3 (6 words)
01A0 0060h to 01A0 0077h	Parameters for event 4 (6 words)
01A0 0078h to 01A0 008Fh	Parameters for event 5 (6 words)
01A0 0090h to 01A0 00A7h	Parameters for event 6 (6 words)
01A0 00A8h to 01A0 00BFh	Parameters for event 7 (6 words)
01A0 00C0h to 01A0 00D7h	Parameters for event 8 (6 words)
01A0 00D8h to 01A0 00EFh	Parameters for event 9 (6 words)
01A0 00F0h to 01A0 0107h	Parameters for event 10 (6 words)
01A0 0108h to 01A0 011Fh	Parameters for event 11 (6 words)
01A0 0120h to 01A0 0137h	Parameters for event 12 (6 words)
01A0 0138h to 01A0 014Fh	Parameters for event 13 (6 words)
01A0 0150h to 01A0 0167h	Parameters for event 14 (6 words)
01A0 0168h to 01A0 017Fh	Parameters for event 15 (6 words)
01A0 0180h to 01A0 0197h	Parameters for event 16 (6 words)
01A0 0198h to 01A0 01AFh	Parameters for event 17 (6 words)
...	...
01A0 05D0h to 01A0 05E7h	Parameters for event 62 (6 words)
01A0 05E8h to 01A0 05FFh	Parameters for event 63 (6 words)
01A0 0600h to 01A0 0617h	Reload/link parameter for Event 0 (6 words)
01A0 0618h to 01A0 062Fh	Reload/link parameter for Event 1 (6 words)
...	...
01A0 07E0h to 01A0 07F7h	Reload/link parameter for Event 20 (6 words)
01A0 07F8h to 01A0 07FFh	Scratch pad area (2 words)

Figure 1−3. EDMA Channel Parameter Entries for Each EDMA Event—C6416 DSP

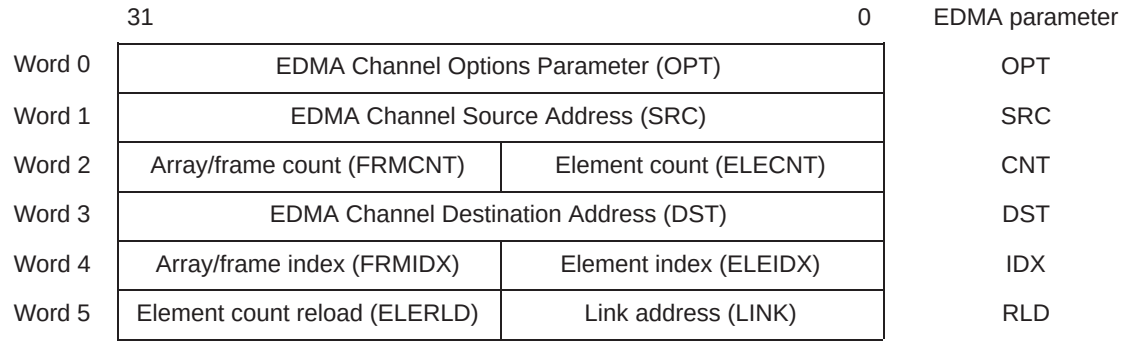


Table 1−3. EDMA Channel Parameter Descriptions—C6416 DSP

Word	Acronym	Parameter
0	OPT	Channel options
1	SRC	Channel source address
2†	ELECNT	Element count
	FRMCNT	Frame count (1D) or Array count (2D)
3	DST	Channel destination address
4†	ELEIDX	Element index
	FRMIDX	Frame index (1D) or Array index (2D)
5†	LINK	Link address
	ELERLD	Element count reload

† Parameter set entries should always be accessed as 32-bit words using the STW or LDW instructions..

1.4 Types of EDMA Transfers

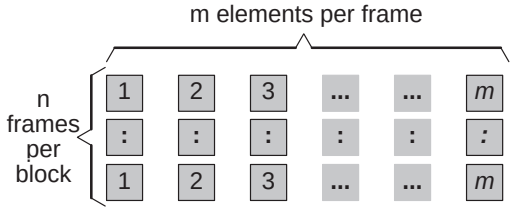
The EDMA provides for two types of data transfers, 1-dimensional (1D) and 2-dimensional (2D). The 2DD and 2DS fields in the channel options parameter register (OPT) select the type of transfer. When the 2DD field is set to 1, a 2D transfer on the destination is performed. Similarly, when the 2DS field is set to 1, a 2D transfer on the source is performed. All combinations of 2DS and 2DD are supported.

The number of dimensions a transfer has will determine the makeup of a frame of data. In a 1D transfer, a number of individual elements make up the frames. In a 2D transfer, blocks are made up of a number of arrays, each of which is made up of a number of elements. For a representation of various types of EDMA transfers, see Appendix A.

1.4.1 1-Dimensional Transfers

For 1D transfers, a group of elements equal to element count constitute a frame. Transfers focus on individual elements. Each frame of data to be transferred has a single dimension associated with it, indicating the number of elements per frame. EDMA channels may be configured to transfer multiple frames (or a block of frames), but each frame is handled individually. Frame count is the number of frames in a 1D transfer. A 1D transfer can be considered two dimensional, with the second dimension fixed at 1. Figure 1−4 shows a sample 1D frame with an element count of *m*.

Figure 1−4. 1-Dimensional Transfer Data Frame



The elements within a block can be all located at the same address, at contiguous addresses, or at a configurable offset from one another. The addresses of elements within a frame can be located at a specific distance apart, as determined by the element index (ELEIDX), while address of the first element of each frame is a set distance from a particular element of the previous frame, as determined by the frame index (FRMIDX). Once a complete frame is transferred, the element count reaches 0. Therefore for multiframe transfers, the element count has to be reloaded by the element count reload field (ELERLD) in the transfer entry.

Transfers may be submitted either one element at a time when element synchronized (FS = 0), or one frame at a time when frame synchronized (FS = 1).

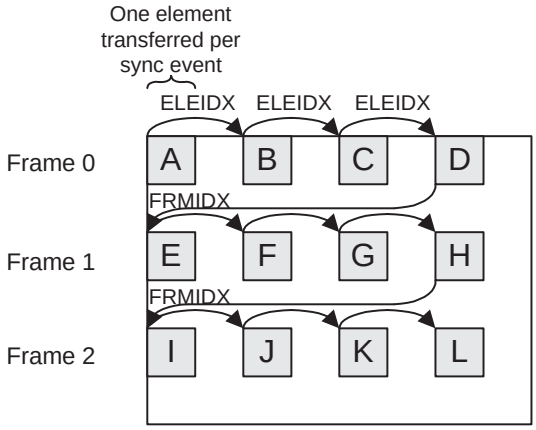
1.4.1.1 Element Synchronized 1D Transfer (FS = 0)

Note:

For element synchronized transfers, ELERLD should be set to the ELECNT value by the programmer.

For an element synchronized transfer, each sync event transfers a single element. After each sync event is received and the single-element TR is submitted to the EDMA, the EDMA channel controller updates the source and destination addresses within the parameter table. Therefore, the element index (ELEIDX) and frame index (FRMIDX) are based on the difference between element addresses. Figure 1−5 shows a 1D element synchronized transfer with 4 elements in each frame (ELECNT = 4) and a total of 3 frames (FRMCNT = 2). In this example, a total of 12 sync events exhaust the PaRAM channel entry.

Figure 1−5. 1D Transfer with Element Synchronization (FS = 0)



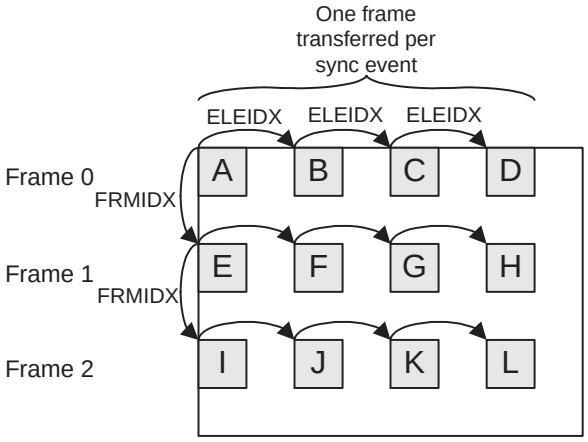
Each element in a frame is transferred from its source to destination address upon receiving the channel-specific sync event. After the channel receives a sync event, it sends off a transfer request for DMA service. The EDMA channel controller then decrements the element count (ELECNT) by 1 in the parameter RAM. When a channel sync event occurs and ELECNT = 1 (indicating the last element in a frame), the EDMA channel controller first sends off the transfer request triggered by the event. Afterward, an element count reload occurs with the 16-bit value in ELERLD and frame count (FRMCNT) decrements by 1. The element index (ELEIDX) is used to compute the address of the next element in a frame. Similarly, the frame index (FRMIDX) is added to the last element address in a frame to derive the next frame start address. The address modification and count modification depends on the type of update modes selected. Specific updates are described in sections 1.9.1 and 1.9.2.

If linking is enabled (LINK = 1, see section 1.11), the complete transfer parameters get reloaded (from the parameter reload space in EDMA channel controller parameter RAM) after sending the last transfer request to the EDMA transfer controller. This sets up a new set of parameters in advance for the next occurrence of the event.

1.4.1.2 Frame Synchronized 1D Transfer (FS = 1)

For a frame synchronized transfer, each sync event transfers a single frame of elements. The frame index no longer represents the difference between the address of the last element of a frame and the address of the first element of the subsequent frame, but rather the difference between the starting addresses of each frame. A frame-synchronized 1D transfer is functionally identical to an array-synchronized 2D transfer (assuming ELEIDX equals the number of bytes per element). Figure 1−6 shows the address indexing for a frame-synchronized 1D transfer with 4 elements in each frame (ELECNT = 4) and a total of 3 frames (FRMCNT = 2). In this example, a total of 3 sync events exhaust the PaRAM channel entry.

Figure 1−6. 1D Transfer With Frame Synchronization (FS = 1)



The element transfer in each frame is not synchronized, but instead the channel event synchronizes each frame transfer. The FS bit, in the channel options parameter register (OPT), should be set to 1 to enable frame-synchronized transfer. The element index (ELEIDX) can be used to stagger elements in a frame. Frame index (FRMIDX) can be added to the start element address in a frame to derive the next frame start address. Element count reload (ELERLD) does not apply to a 1D frame-synchronized transfer (FS = 1). The address modification and count modification depends on the type of update modes selected. Sections 1.9.1 and 1.9.2 describe specific updates.

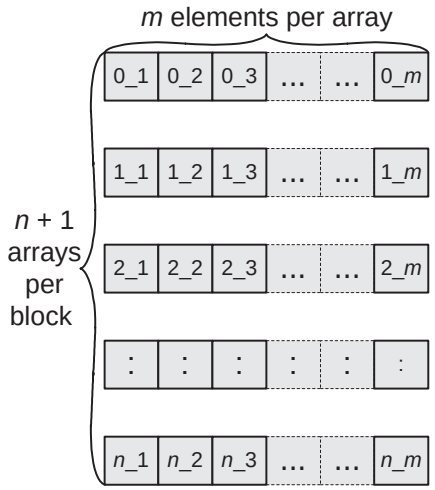
If linking is enabled (LINK = 1, see section 1.11), the complete transfer parameters get reloaded (from the parameter reload space in EDMA channel controller parameter RAM) after sending the last transfer request to the EDMA transfer controller.

1.4.2 2-Dimensional Transfers

2D transfers are useful for imaging applications where a contiguous set of elements (referred to as array) has to be transferred on receiving a sync event. This means there is no spacing or indexing between elements in an array, hence 2D transfers do not use the element index (ELEIDX). The number of elements in an array makes up for the first dimension of the transfer. A group of arrays forms the second dimension, called a block. Arrays can be offset from one another by a fixed amount. Figure 1−7 shows a 2D frame with an array count of *n* and an element count of *m*.

The offset of the arrays is determined by the array index (FRMIDX), the value of which depends on the synchronization mode of the transfer. Transfers may be submitted either one array at a time when array synchronized (FS = 0), or one block at a time when block synchronized (FS = 1).

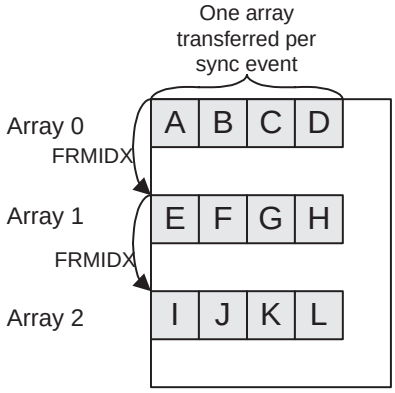
Figure 1−7. 2-Dimensional Transfer Data Block



1.4.2.1 Array Synchronized 2D Transfer (FS = 0)

For an array synchronized transfer, each sync event transfers a single array of contiguous elements. A channel that is configured to perform a 2D transfer with array synchronization updates its source and destination registers after the transfer request for each array is submitted. The array index (FRMIDX) is the difference between the starting addresses for each array of the block. Figure 1−8 shows an array-synchronized 2D transfer with 4 elements in each frame (ELECNT = 4) and a total of 3 arrays (FRMCNT = 2). In this example, a total of 3 sync events exhaust the PaRAM channel entry. FRMIDX is used for all address update modes except fixed address update mode (SUM/DUM = 00b).

Figure 1−8. 2D Transfer with Array Synchronization (FS = 0)



Upon receiving a synchronization event, an array (contiguous group of elements) is transferred. Figure 1−8 shows 4 elements in an array (ELECNT = 4) and the number of arrays to be transferred is 3 (FRMCNT = 2). The frame count (FRMCNT) decrements after the transfer of each array. The frame index is added to an array's start address to derive the next array's start address. The actual address modification and count modification depends upon the type of update modes selected. Sections 1.9.2 and 1.11 describe specific updates.

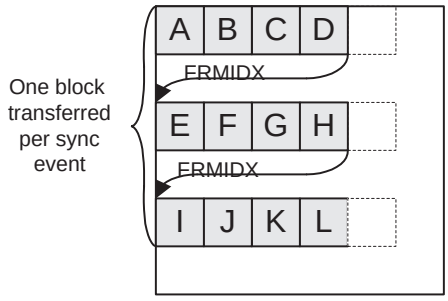
When FRMCNT reaches 0 and linking is enabled (LINK = 1, see section 1.11), the complete transfer parameters get reloaded (from the parameter reload space in EDMA channel controller parameter RAM) after sending the last transfer request to the address generation hardware.

1.4.2.2 Block Synchronized 2D Transfer (FS = 1)

For a 2D block synchronized transfer, a single sync event transfers an entire block of arrays. For a 2D transfer, the complete block gets transferred when the channel's event occurs and FS = 1. Block synchronization causes the address generation/transfer logic to implement the array index (FRMIDX). This address update is transparent and is not reflected in the parameter RAM. The address is updated after each element in a burst. The logic first updates the addresses according to the setting of SUM/DUM. If an element is the last in a particular array and an update mode is selected (SUM/DUM ≠ 00b), the address(es) are indexed according to the array index. The index is added to the address after the address update occurs. FRMIDX is equal to the space between arrays of a block, as shown in Figure 1−9.

If linking is enabled (LINK = 1), the next EDMA block transfer in the link (as specified by the link address) is performed as soon as the next block sync arrives. See section 1.11 for details on linking.

Figure 1−9. 2D Transfer with Block Synchronization (FS = 1)



1.5 Initiating an EDMA Channel Controller Transfer

There are three ways to initiate a programmed data transfer using the EDMA channel controller (EDMACC):

- Event-triggered transfer request (this is a typical usage of the EDMACC). Allows for a peripheral, system, or externally-generated event to trigger a transfer request (TR).
- Chain-triggered transfer request. Allows for completion of a given transfer to trigger a new EDMA channel to submit a transfer request (TR).
- CPU-triggered transfer request. Allows the CPU to manually trigger a transfer request (TR) by writing to the event set register (ESR) for a given channel.

Quick-DMA (QDMA) transfers provide a way to submit a transfer request directly to the EDMA transfer controller, bypassing the EDMA channel controller. As this is not part of the EDMA channel controller, this is described in section 1.16.

For all TRs, regardless of the mode of initiation, the EDMA transfer controller queues the request and executes the data transfer when the TR reaches the head of the programmed EDMA transfer controller priority queue.

1.5.1 Event-Triggered Transfer Request

When an event is asserted, that event gets latched in the corresponding channel of the event register (ER*n* = 1). If the corresponding channel of the event enable register is enabled (EER*n* = 1), then the EDMA channel controller processes the corresponding PaRAM channel entry. If the PaRAM channel entry is a NULL entry, then no transfer request (TR) is submitted and the corresponding ER*n* bit is cleared. If the PaRAM channel is a valid entry (not a NULL entry), then a TR is submitted to the EDMA transfer controller and the corresponding ER*n* bit is cleared. Once the TR is submitted and the ER*n* bit is cleared (not necessarily when the data transfer has occurred), a new event can be received/processed by the EDMA channel controller.

When an event is received, the corresponding bit in the event register (ER*n*) is set, regardless of the state of EER*n*. If the event is disabled when an event is received (ER*n* = 1 and EER*n* = 0), the ER*n* bit remains set. If the event is subsequently enabled (EER*n* = 1), then a TR will be submitted and the ER*n* bit will be cleared.

1.5.2 Chain-Triggered Transfer Request

Chaining is a mechanism by which the completion of one transfer automatically triggers the transfer request (TR) submission for another channel. When a transfer completion code (TCC/ATCC) is received, if the corresponding bit in the channel chain enable register is enabled (CCER*n* = 1), then the EDMA channel controller processes the corresponding PaRAM channel entry. If the PaRAM channel entry is a NULL entry, then no TR is submitted. If the PaRAM channel entry is a valid entry (not a NULL entry), then a TR is submitted to the EDMA transfer controller. After the TR is submitted (not necessarily when the transfer occurs), a new TCC can be received/processed by the EDMA channel controller.

Transfer completion functionality is shared between chaining and interrupt generation. When a TCC is received, the corresponding bit in the channel interrupt pending register is set (CIP*n* = 1), regardless of the state of either channel interrupt enable register (CIER) or channel chain enable register (CCER). If the channel is subsequently enabled for chaining (CCER*n* = 1), then a TR will not be submitted and the CIPR*n* bit remains set. A new TCC must be received for a TR to be submitted. See section 3.3 or section 4.3 for chaining details.

1.5.3 CPU-Triggered Transfer Request

The CPU can write to the event set register (ESR) to start an EDMA transfer. Writing a 1 to the corresponding event in ESR triggers a transfer request (TR) submission, regardless of the state of the EER*n* bit. Just as with an event-triggered or chain-triggered transfer, the transfer parameters in the EDMA channel controller PaRAM are evaluated and a TR is submitted to the EDMA transfer controller if the entry is not a null entry. The corresponding enable bit does not have to be set in the event enable register (EER) for CPU-initiated EDMA transfers.

1.6 EDMA Channel Controller Event-to-Channel Mapping

All EDMA channels are tied to a specific synchronization event. Synchronization allows EDMA transfers to be triggered by events from peripherals, external hardware, or an EDMA transfer completion event. A channel only requests a data transfer when it receives its event or when the CPU manually synchronizes it (by writing to ESR). The amount of data to be transferred depends on the channel's configuration. A channel can submit an entire frame/block when frame/block-synchronized, or a subset of a frame (element or array, depending on dimension) when element/array-synchronized.

Table 1−4 and Table 1−5 give examples of the synchronization events associated with each of the programmable EDMA channels for the C621x/C671x DSP and C64x DSP, respectively.

On most C6000 EDMA-based devices, the association of an event to a channel is fixed. Each of the EDMA channels has one specific event associated with it. For example, on C6211 DSP (see Table 1−4) timer interrupt 0 (TINT0) is always associated with EDMA channel 1.

Some C6000 devices (C6713 DSP, for example) include an event selector block that enables you to select which events are associated with a given EDMA channel.

Refer to the device-specific data manual to determine specific event-to-channel mappings, and to determine whether your device includes an event selector.

If a specified channel is not event-triggered (for example, channel 0 is not triggered by DSPINT), that PaRAM entry can be used for CPU-triggered or chain-triggered transfers, or for linking.

Table 1−4. EDMA Channel Synchronization Events—C6211 DSP

EDMA Channel Number†	Event Acronym	Event Description
0	DSPINT	Host port to DSP interrupt
1	TINT0	Timer 0 interrupt
2	TINT1	Timer 1 interrupt
3	SD_INT	EMIF SDRAM timer interrupt
4	EXT_INT4	External interrupt pin 4
5	EXT_INT5	External interrupt pin 5
6	EXT_INT6	External interrupt pin 6
7	EXT_INT7	External interrupt pin 7
8	EDMA_TCC8	EDMA transfer complete code 1000b interrupt
9	EDMA_TCC9	EDMA transfer complete code 1001b interrupt
10	EDMA_TCC10	EDMA transfer complete code 1010b interrupt
11	EDMA_TCC11	EDMA transfer complete code 1011b interrupt
12	XEVT0	McBSP0 transmit event
13	REVT0	McBSP0 receive event
14	XEVT1	McBSP1 transmit event
15	REVT1	McBSP1 receive event

† EDMA channels 8 to 11 are used for transfer chaining only. See section 3.3, *Chaining EDMA Channels by an Event*.

Table 1−5. EDMA Channel Synchronization Events—C6416 DSP

EDMA Channel Number†	Event Acronym	Event Description
0	DSPINT	Host port to DSP interrupt
1	TINT0	Timer 0 interrupt
2	TINT1	Timer 1 interrupt
3	SD_INT0	EMIFA SDRAM timer interrupt
4	GPINT4/EXT_INT4	GPIO event 4/External interrupt 4
5	GPINT5/EXT_INT5	GPIO event 5/External interrupt 5
6	GPINT6/EXT_INT6	GPIO event 6/External interrupt 6
7	GPINT7/EXT_INT7	GPIO event 7/External interrupt 7
8	GPINT0	GPIO event 0
9	GPINT1	GPIO event 1
10	GPINT2	GPIO event 2
11	GPINT3	GPIO event 3
12	XEVT0	McBSP 0 transmit event
13	REVT0	McBSP 0 receive event
14	XEVT1	McBSP 1 transmit event
15	REVT1	McBSP 1 receive event
16	−	None
17	XEVT2	McBSP 2 transmit event
18	REVT2	McBSP 2 receive event
19	TINT2	Timer 2 interrupt
20	SD_INT1	EMIFB SDRAM timer interrupt
21	PCI	PCI wakeup interrupt
22−27	−	None
28	VCPREVT	VCP receive interrupt

† Each of the 64 channels can also be synchronized with the transfer completion or alternate transfer completion events. See section 4.3, *Chaining EDMA Channels by an Event*.

Table 1−5. EDMA Channel Synchronization Events—C6416 DSP (Continued)

EDMA Channel Number†	Event Acronym	Event Description
29	VCPXEVT	VCP transmit interrupt
30	TCPREVT	TCP receive interrupt
31	TCPXEVT	TCP transmit interrupt
32	UREVT	UTOPIA receive event
33−39	−	None
40	UXEVT	UTOPIA transmit event
41−47	−	None
48	GPINT8	GPIO event 8
49	GPINT9	GPIO event 9
50	GPINT10	GPIO event 10
51	GPINT11	GPIO event 11
52	GPINT12	GPIO event 12
53	GPINT13	GPIO event 13
54	GPINT14	GPIO event 14
55	GPINT15	GPIO event 15
56−63	−	None

† Each of the 64 channels can also be synchronized with the transfer completion or alternate transfer completion events. See section 4.3, *Chaining EDMA Channels by an Event*.

1.7 Element Size and Alignment

The element size that the EDMA channel controller (EDMACC) uses for a transfer is specified in the ESIZE field of the channel op†tions parameter register (OPT). The EDMACC can transfer 32-bit words, 16-bit half-words, or 8-bit bytes in a transfer. The EDMA transfers have higher throughput when using 32-bit words; 16-bit and 8-bit transfers always result in lower throughput.

The addresses must be aligned on the element size boundary. Word accesses must be aligned on a word (multiple of 4) boundary and half-word accesses must be aligned on a half-word (multiple of 2) boundary. Unaligned values will result in undefined operation.

1.8 Fixed-Mode Transfer Considerations

The maximum EDMA element size is a 32-bit word; however, the following EDMA transfer controller data paths are 64-bit wide:

- L2 SRAM (all C6000 EDMA-based devices)
- EMIFA (interfaced to 64-bit-wide memory, C64x DSP only)
- TCP and VCP (6416 DSP only)
- Video port (DM642 DSP and C6412 DSP)
- Refer to your device-specific data manual for additional examples

When transferring a burst of elements to or from a 64-bit-wide peripheral (for example, L2 SRAM or EMIFA), 64-bit elements are transferred to maximize the available bandwidth if the element size is a 32-bit word (ESIZE = 00b). Because of this optimization, care must be taken when performing a fixed-mode access (SUM or DUM = fixed) to peripherals that have 64-bit data paths to/from the EDMA transfer controller.

- If the EDMA is setup with the following parameters:
  - Element size is 32-bit word (ESIZE = 00b in OPTIONS field)
  - Fixed-address mode on either source or destination (SUM or DUM = 00b in OPTIONS field)
  - Transfer/synchronization type is array/frame/block-synchronized (not element-synchronized, see section 1.9.1)
  - Element count is greater than 1 (ELECNT > 1)
  - Either the source or destination bus width is 64 bits

- Then you must ensure that the following conditions are true:
  - Element count (ELECNT) must be a multiple of 2
  - Frame/Array index field must be a multiple of 8 bytes (2 words, 1 doubleword)
  - Both the source address and destination address must be double-word aligned (that is, a multiple of 8 bytes (2 words, 1 doubleword))

  Operation is undefined, if the above conditions are not met.

Accesses to a 64-bit-wide data bus with the above EDMA configurations are fixed on a 64-bit boundary and always perform doubleword accesses. For example, when performing N number of 32-bit accesses to L2 SRAM or a 64-bit device on EMIFA in fixed-address mode (ELECNT = N, N > 1), the EDMA transfer controller actually performs N/2 number of 64-bit accesses to the fixed doubleword address. Thus, it is actually a 64-bit doubleword that is transferred.

Assuming the above conditions with the destination specified as fixed (DUM = 00b), both word 0 and word 1 of the fixed doubleword destination address are updated. For example, a fixed-mode, 8-element, 32-bit ESIZE write to a 64-bit EMIF at byte address 8000 0000h updates both word 0 (at address 8000 0000h) and word 1 (at address 8000 0004h) with the new data in a single EMIF bus cycle. This is repeated for a total of 4 times. Thus, if the EMIF is interfaced to a 64-bit FIFO, the full 64-bit bus width is utilized.

Assuming the above conditions with the source specified as fixed (SUM = 00b), both word 0 and word 1 of the fixed doubleword source address are extracted. For example, a fixed-mode, 8-element, 32-bit ESIZE read from a 64-bit EMIF at byte address 8000 0000h results in word 0 (at address 8000 0000h) and word 1 (at address 8000 0004h) being read in a single cycle. This is repeated for a total of 4 times. Thus, if the EMIF is interfaced to a 64-bit FIFO, the full 64-bit bus width is utilized. Otherwise, if the EMIF (or L2) is being used as a memory and a fill operation is being performed (fixed source to incrementing destination address, for example), you must ensure that both word 0 and word 1 of the doubleword-aligned address are set with the desired fill value.

1.9 Parameter Set Updates

1.9.1 Element and Frame/Array Count Updates

The EDMA channel controller parameter entry contains values for the element count (ELECNT) and frame/array count (FRMCNT). Each entry is a 16-bit unsigned value. ELECNT specifies the actual number of elements in a frame or array. The maximum number of elements in a frame or an array is 65535. FRMCNT specifies the actual number of frames/arrays minus 1. The maximum number of frames or arrays in a block is 65536 (since FRMCNT of 0 implies 1, 1 implies 2, ..., 65535 implies 65536.)

The EDMA channel controller updates the ELECNT and FRMCNT in the corresponding channels transfer entry after each transfer request is submitted, depending on the synchronization type of the transfer (based on the 2DS/2DD and FS bits in OPTIONS field) as shown in Table 1−6. In general, the EDMA channel controller must track the ELECNT and FRMCNT fields that are necessary to correctly track the number of synchronization events/transfer requests that compose a complete parameter set. The ELECNT and FRMCNT fields that are constant for a given transfer request type are tracked by the EDMA transfer controller, and do not need to be tracked by the EDMA channel controller.

Table 1−6. EDMA Element and Frame/Array Count Updates

Synchronization	Transfer Mode	Element Count Update	Frame/Array Count Update†
Element Sync	1D (FS = 0, 2DS & 2DD = 0)	−1 (reload if ELECNT = 1)	−1 (if element count = 1)
Frame Sync	1D (FS = 1, 2DS & 2DD = 0)	None	−1
Array Sync	2D (FS = 0, 2DS | 2DD = 1)	None	−1
Block Sync	2D (FS = 1, 2DS | 2DD = 1)	None	None

† Frame count update applies to 1D transfers. Array count update applies to 2D transfers. No frame/array count update occurs if the frame/array count is 0 (FRMCNT = 0).

Reloading the element count for element synchronized transfers has a special condition. When a sync event occurs at the end of a frame (ELECNT = 1 prior to event), the EDMA channel controller sends off the transfer request and reloads ELECNT from the element count reload field in the parameter RAM. This element count reload occurs when element count is 1 and the frame count is nonzero (that is, there are additional frames to transfer). When configuring an element synchronized transfer, the ELERLD field should always be set to the initial value of the ELECNT field to ensure proper operation.

1.9.2 Source/Destination Address Updates

The EDMA channel controller updates the source address (SRCADDR) and destination address (DSTADDR) in the corresponding channels parameter set after each transfer request is submitted, depending on the synchronization type of the transfer (based on the 2DS, 2DD, and FS bits of the OPTIONS field, as shown in Table 1−6). The EDMA transfer controller performs the address updates within a single transfer request according to the source address update mode (SUM) and destination address update mode (DUM) bits, as described in Table 1−7. The various address update modes listed in Table 1−7 provide for the creation of a variety of data structures. The source and/or destination address is updated depending on whether frame/block sync (FS) is enabled, or the dimension (2DS/2DD) of the transfer. All address updates occur after the current transfer request is sent; therefore, these updates are used to set the EDMA channel controller parameters for the next event.

The update of the source and/or destination address depends on the transfer type chosen for both the source and destination. For example, an array synchronized transfer results in the EDMA channel controller updating both the source address and the destination address according to the FRMIDX or ELECNT of the transfer. The EDMA transfer controller performs address updates within the array based on the SUM/DUM bit as reads/writes are performed to the ports. Table 1−8 (page 1-28) shows the amount of modification of the source address for each of the combinations of frame sync (FS), transfer type (2DS/2DD), and SUM values. Table 1−9 (page 1-29) shows the possible destination address updates.

Note that when either the source or the destination is a 2D transfer and the transfer is block synchronized (FS = 1), the complete block of data is transferred on a sync event. Therefore, EDMA channel controller address updates are not applicable in this case because the entire parameter set is submitted as a single transfer request and the EDMA transfer controller manages the address updates (that are not visible in the PaRAM). If LINK = 1 and the link conditions outlined in Table 1−10 are met, the link parameters are copied directly to the event parameter.

Table 1−7. Source/Destination Address Update Modes

SUM/DUM Bit Value (Binary)	Address Modification	1D Transfer	2D Transfer
00	None	All elements located at the same address.	All elements in an array are at the same address.
01	Increment	All elements are contiguous, with subsequent elements located at a higher address than the previous.	All elements within an array are contiguous, with subsequent elements located at a higher address than the previous. Arrays are offset by FRMIDX.
10	Decrement	All elements are contiguous, with subsequent elements located at a lower address than the previous.	All elements within an array are contiguous, with subsequent elements located at a lower address than the previous. Arrays are offset by FRMIDX.
11	Index	All elements within a frame are offset from one another by ELEIDX. Frames are offset by FRMIDX.	Reserved.

Table 1−8. EDMA Source Address Parameter Updates

Sync Type	FS	2DS:2DD	Source Update Mode (SUM) 00	01	10	11
Element	0	00	None	+ESIZE Increment by element size.	−ESIZE Decrement by element size.	+ELEIDX or + FRMIDX if ELECNT = 1 Add signed ELEIDX to each element in a frame except the last. Add signed FRMIDX to the last element in a frame when ELECNT = 1.
Array	0	01	None	+(ELECNT × ESIZE bytes) Add ELECNT scaled by element size to the start address of previous frame.	−(ELECNT × ESIZE bytes) Subtract ELECNT scaled by element size from the start address of previous frame.	Reserved
		10	None	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame are in increasing order.	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame are in decreasing order.	Reserved
		11	None	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame are in increasing order.	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame are in decreasing order.	Reserved
Frame	1	00	None	+(ELECNT × ESIZE bytes) Add ELECNT scaled by element size to the start address of previous frame.	−(ELECNT × ESIZE bytes) Subtract ELECNT scaled by element size from the start address of previous frame.	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame spaced by ELEIDX.
Block	1	01	None	None	None	Reserved
		10	None	None	None	Reserved
		11	None	None	None	Reserved

Legend: ELECNT: Element count; ELEIDX: 16-bit signed element index value; FRMCNT: Frame/array count; FRMIDX: 16-bit signed frame index value(1D transfers) or 16-bit signed array index value (2D transfers); ESIZE: element size in bytes

Table 1−9. EDMA Destination Address Parameter Updates

Sync Type	FS	2DS:2DD	Destination Update Mode (DUM) 00	01	10	11
Element	0	00	None	+ESIZE Increment by element size.	−ESIZE Decrement by element size.	+ELEIDX or + FRMIDX if ELECNT = 1 Add signed ELEIDX to each element in a frame except the last. Add signed FRMIDX to the last element in a frame when ELECNT = 1.
Array	0	01	None	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame are in increasing order.	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame are in decreasing order.	Reserved
		10	None	+(ELECNT × ESIZE bytes) Add ELECNT scaled by element size to the start address of previous frame.	−(ELECNT × ESIZE bytes) Subtract ELECNT scaled by element size from the start address of previous frame.	Reserved
		11	None	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame are in increasing order.	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame are in decreasing order.	Reserved
Frame	1	00	None	+(ELECNT × ESIZE bytes) Add ELECNT scaled by element size to the start address of previous frame.	−(ELECNT × ESIZE bytes) Subtract ELECNT scaled by element size from the start address of previous frame.	+FRMIDX Add signed FRMIDX to the first element in a frame. Element addresses in a frame spaced by ELEIDX.
Block	1	01	None	None	None	Reserved
		10	None	None	None	Reserved
		11	None	None	None	Reserved

Legend: ELECNT: Element count; ELEIDX: 16-bit signed element index value; FRMCNT: Frame/array count; FRMIDX: 16-bit signed frame index value(1D transfers) or 16-bit signed array index value (2D transfers); ESIZE: element size in bytes

1.10 Channel Completion Conditions

A parameter set for a given channel is complete when the required number of transfer requests are submitted (based on receiving the required number of synchronization events). This is shown in Table 1−10 for each transfer type.

Table 1−10. Channel Completion Conditions For Transfer Type

Sync type	Transfer request size per sync event	Total number of sync events for channel completion	Parameter set state prior to last sync event
Element Sync	1	ELECNT × (FRMCNT + 1)	FRMCNT = 0 && ELECNT = 1
Frame Sync	ELECNT	FRMCNT + 1	FRMCNT = 0
Array Sync	ELECNT	FRMCNT + 1	FRMCNT = 0
Block Sync	ELECNT × (FRMCNT + 1)	1	Any

The EDMA channel controller handles two distinct operations upon completion of a transfer request.

- First, if a channel is enabled to generate a transfer completion interrupt (TCINT = 1), the TR is submitted with a request to the EDMA transfer controller to signal completion along with the user-programmed transfer completion code (TCC). When the transfer is complete within the transfer controller, the specific TCC is sent to the EDMA channel controller. The EDMA channel controller uses this code to trigger another EDMA channel to occur (chaining, refer to section 3.3 or section 4.3) or to generate an interrupt to the CPU. (Note that the C64x EDMA generates an interrupt based on completion as described here or for every transfer request, alternate completion.)
- Second, if a channel is enabled for linking (LINK = 1) , the link update is performed after the final TR is submitted. This prepares the channel's parameter entry for the next synchronization event. This is described in section 1.11.

1.11 Linking EDMA Transfers

The EDMA channel controller (EDMACC) provides linking, a feature especially useful for maintaining ping-pong buffers, complex sorting, and circular buffering all with no CPU intervention. If LINK = 1, upon completion of a transfer, the EDMACC link feature reloads the current transfer parameters with the parameter pointed to by the 16-bit link address. The entire EDMACC parameter RAM is located in the 01A0 xxxxh area. Therefore, the 16-bit link address, which corresponds to the lower 16-bit physical address, suffices to specify the location of the next transfer entry. The link address must be aligned on a 24-byte boundary. Figure 1−10 shows an example of a linked EDMACC transfer.

Note:

All enabled EDMA channel parameter sets must enable linking for proper operation. The parameter should either link to another useful transfer or to NULL. Refer to section 1.12.

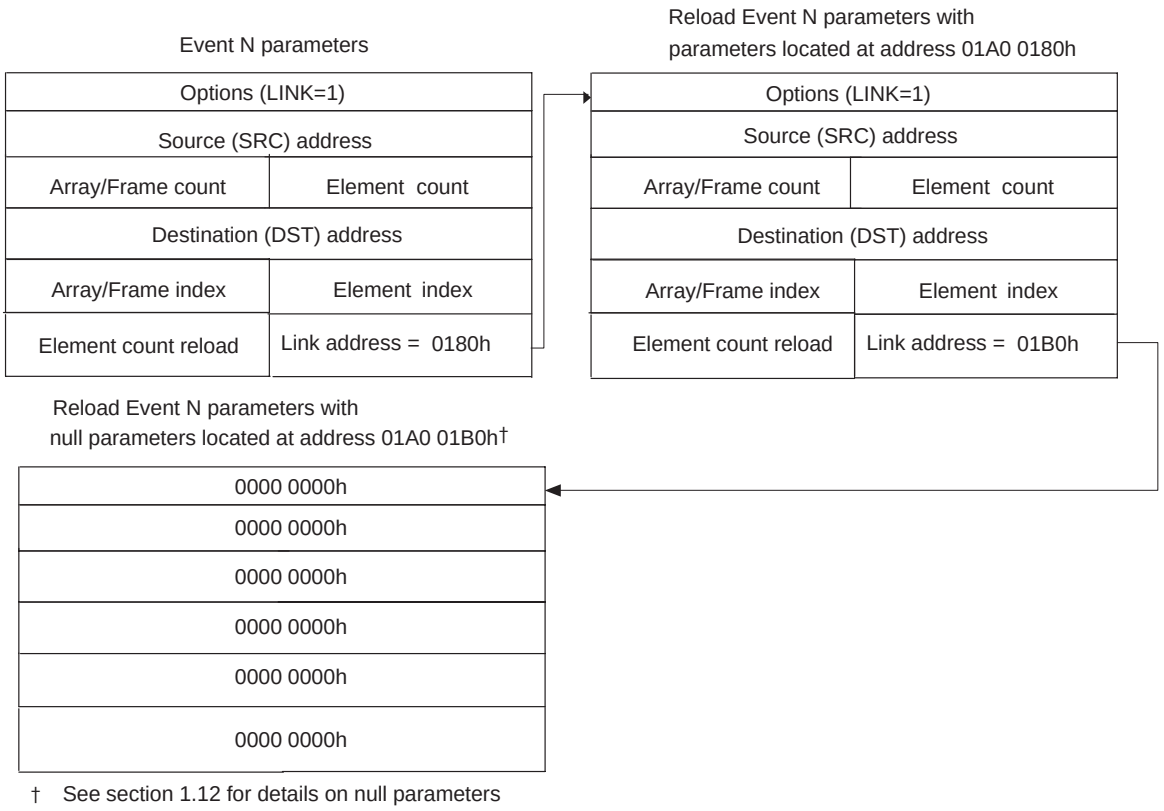
The link address is evaluated only if LINK is set to 1 and only after the event parameters have been exhausted. An event's parameters are exhausted when the EDMACC has completed the transfer associated with the request. Table 1−10 shows the channel completion conditions when the linking of parameters is performed. There is virtually no limit to the length of linked transfers. The last transfer parameter entry should have its LINK = 1 to link to a NULL parameter set so that the linked transfer stops after the last transfer. See section 1.12 for details.

Linking an entry to itself replicates the behavior of autoinitialization to facilitate the use of circular buffering and repetitive transfers. After an EDMA channel exhausts its current entry, it reloads the parameter set and the transfer begins again.

Once the channel completion conditions are met for an event, the transfer parameters located at the link address are loaded into one of the 16 event parameter space (C621x/C671x DSP) or 64 event parameter space (C64x DSP) for the corresponding event. Now, the EDMACC can start the next transfer. To eliminate possible timing windows posed during this parameter reload mechanism, the EDMACC does not evaluate the event register during this time. However, the event register still captures events, and processes them after the parameter reload is complete.

Any entry in the PaRAM can be used for a linked transfer parameter set. Entries in the first 16 (C621x/C671x DSP) or 64 (C64x DSP) locations should only be used for linking, if the corresponding event and chain event are disabled.

Figure 1−10. Linked EDMA Transfer

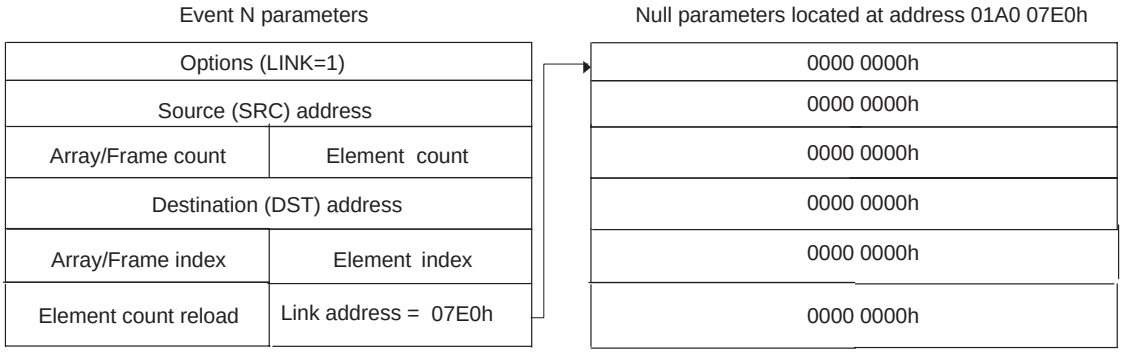


† See section 1.12 for details on null parameters

1.12 Terminating an EDMA Transfer

All EDMA channel controller (EDMACC) transfers must terminate by linking to a NULL parameter set after the last transfer. The NULL parameter set serves as the termination point of any EDMA transfer. A NULL parameter set is defined as an EDMACC parameter set where all the parameters (options, source/destination address, frame/element count, etc.) are cleared to 0. Multiple EDMA transfers can link to the same terminating NULL parameter set. Therefore, the EDMACC parameter RAM only requires one NULL parameter set. Figure 1−11 presents an example of an EDMA transfer termination.

Figure 1−11. Terminating EDMA Transfers



1.13 Chaining EDMA Channels

The channel chaining capability for the EDMA channel controller (EDMACC) allows the completion of an EDMA channel transfer to trigger another EDMA channel transfer.

Chaining is different from linking (section 1.11). The EDMACC link feature reloads the current channel parameter set with the linked parameter set. The EDMACC chaining feature does not modify or update any channel parameter set, it provides a synchronization event to the chained channel. See section 1.5.2.

1.14 Transfer Complete Code (TCC) Generation

The EDMA channel controller (EDMACC) and EDMA transfer controller (EDMATC) provide a user-programmable transfer completion code (TCC) that is programmable in the TCC field of the channel options parameter (OPT). If enabled for completion code generation (TCINT = 1) on the final TR submission for a parameter set, the EDMACC submits the transfer completion code to the EDMATC along with the TR (see Table 1−10 for details on when a parameter set is complete, and thus when the last TR is submitted). When the EDMATC has successfully transferred all the data for the final TR, the EDMATC sends the completion code back to the EDMACC to inform the EDMACC that the transfer is complete.

In summary, the EDMACC submits the user-programmed TCC code (value of *n*) to the EDMATC when the following are true:

- Final TR for a channel parameter set is submitted to the EDMATC
- TCINT = 1, in OPT
- TCC = *n*, in OPT

When the EDMACC receives transfer completion status back from the EDMATC, the TCC code can cause:

- Interrupt generation via EDMA_INT
  - CIPR*n* bit is set to 1
  - EDMA_INT is asserted, if CIER*n* bit is enabled
- Chained-trigger synchronization
  - TR is submitted for channel *n*, if CCER*n* is enabled

There is no relationship between the TCC value of *n* and the channel number for the channel being programmed. The value of *n* affects which bit in CIPR is set for EDMA_INT generation and/or which channel is chain-triggered.

The completion code is sent from the EDMATC to the EDMACC after all the data has been transferred for that TRP. This prevents race conditions since when the interrupt is asserted, this assures that the data has reached the destination in memory.

In the C621x/C671x 16-channel EDMA, the TCC field can be programmed with a value between 0 and 15. In the C64x 64-channel EDMA, the TCC field can be programmed to a value between 0 and 63. The 6-bit transfer complete code of the C64x EDMA is the concatenation of the TCCM bits (2 most-significant bits of the transfer complete code) and the TCC bits (4 least-significant bits of the completion code). The transfer complete code is directly mapped to the CIPR bits as shown in Table 1−11 for the C621x/C671x DSP and as shown in Table 1−12 for the C64x DSP.

For example, if TCC = 1100b (and also TCCM = 00 for the C64x DSP), CIPR[12] (C621x/C671x DSP) or CIPRL[12] (C64x DSP) is set to 1 after the transfer is complete, and this generates a CPU interrupt only if CIER[12] = 1. You can program the transfer complete code to any value in Table 1−11 for any EDMA channel. In other words, there does not need to be a direct relation between the channel number and the transfer complete code value. This allows multiple channels having the same transfer complete code value to cause the CPU to execute the same ISR (for different channels). Alternatively, the same channel can set multiple complete codes depending on the transfers performed.

Table 1−11. Transfer Complete Code (TCC) to EDMA Interrupt Mapping (C621x/C671x DSP)

TCC Bits in OPT (TCINT = 1)	CIPR Bit Set	TCC Bits in OPT (TCINT = 1)	CIPR Bit Set
0000b	CIP0	1000b	CIP8
0001b	CIP1	1001b	CIP9
0010b	CIP2	1010b	CIP10
0011b	CIP3	1011b	CIP11
0100b	CIP4	1100b	CIP12
0101b	CIP5	1101b	CIP13
0110b	CIP6	1110b	CIP14
0111b	CIP7	1111b	CIP15

Table 1−12. Transfer Complete Code (TCC) to EDMA Interrupt Mapping (C64x DSP)

TCC Bits in OPT (TCINT = 1)	CIPRL Bit Set	TCC Bits in OPT (TCINT = 1)	CIPRH Bit Set†
00 0000b	CIP0	10 0000b	CIP32
00 0001b	CIP1	10 0001b	CIP33
00 0010b	CIP2	10 0010b	CIP34
00 0011b	CIP3	10 0011b	CIP35
00 0100b	CIP4	10 0100b	CIP36
...	...	...	...
...	...	...	...
01 1110b	CIP30	11 1110b	CIP62
01 1111b	CIP31	11 1111b	CIP63

† Bit fields CIP[32−63] correspond to bits 0 to 31 in CIPRH.

1.14.1 Alternate Transfer Complete Code (ATCC) Generation (C64x EDMA only)

The C64x EDMA also supports (in addition to TCC) a user-programmable alternate transfer complete code (ATCC) that is programmable in the ATCC field of the channel options parameters (OPT). If enabled for alternate transfer complete code generation (ATCINT = 1), on every TR submission except for the final TR of a parameter set, the EDMACC submits the alternate transfer complete code to the EDMATC along with the TR (see Table 1−13 for details on when the a parameter set is complete, and thus when the last TR is submitted). When the EDMATC has successfully transferred all the data for the TR, the EDMATC sends the completion code back to the EDMACC to inform the EDMACC that the transfer is complete.

In summary, the EDMACC submits the user-programmed ATCC code (value of *n*) to the TC when the following are true:

- Any TR except the final TR for a channel parameter set is submitted to the EDMATC
- ATCINT = 1, in OPT
- ATCC = *n*, in OPT

Just as for normal completion via TCC, when the EDMACC receives alternate transfer completion status back from the EDMATC, the ATCC code can cause:

- Interrupt generation via EDMA_INT
  - CIPR*n* bit is set to 1
  - EDMA_INT is asserted, if CIER*n* bit is enabled
- Chained-trigger synchronization
  - TR is submitted for channel *n*, if CCER*n* is enabled

For example in a 1D element-synchronized transfer, alternate transfer complete interrupt may be generated upon transfer completion of each element. The total number of alternate transfer complete codes (ATCC) generated is equal to the total number of sync events or transfer requests (TR) minus 1; that is, an ATCC is signaled for every sync event or TR except for the final event or TR that generates the combined transfer complete code (TCC). See Table 1−13.

Two new fields, the alternate transfer complete interrupt (ATCINT) and the alternate transfer complete code (ATCC), are added to the channel options parameters (OPT). The function of the alternate transfer code is similar to the function of the transfer complete code. Similar to the TCCM:TCC, the ATCC can be set to any values between 0−63 (see Table 1−12). TCC and ATCC can be set to the same value or to a different value relative to one another.

Table 1−13. Channel Completion Conditions For Alternate Transfer Complete Codes

Sync Type	Total number of alternate transfer complete code (ATCC) requests	Total number of completion code requests
Element Sync	1st ((ELECNT × (FRMCNT + 1)) − 1) events	1 for last event/TR
Frame Sync	1st FRMCNT events	1 for last event/TR
Array sync	1st FRMCNT events	1 for last event/TR
Block sync	0	1 for last event/TR

† Note that actual number of frames = FRMCNT + 1

1.15 EDMA Interrupt Generation Based on Completion Code

Unlike the C620x/C670x DMA controller, which has individual interrupts for each DMA channel, the EDMA generates a single interrupt (EDMA_INT) to the CPU on behalf of all 16 channels (C621x/C671x DSP) or 64 channels (C64x DSP).

When the EDMA channel controller (EDMACC) receives a completion code (TCC or ATCC) of value *n* from the EDMA transfer controller (EDMATC), which informs the EDMACC that a previously submitted TR with TCC or ATCC programmed to *n* has completed, the EDMACC sets bit *n* in the channel interrupt pending register (CIPR). The EDMA_INT to the CPU is generated, if the corresponding interrupt enable bit is set in the channel interrupt enable register (CIER).

Note that the C64x EDMA has two channel interrupt pending registers to accommodate the 64 channels: channel interrupt pending low register (CIPRL) for n = 31 to 0 and channel interrupt pending high register (CIPRH) for n = 63 to 32). Also, the C64x EDMA has two channel interrupt enable registers: channel interrupt enable low register (CIERL) and channel interrupt enable high register (CIERH).

CIPR is equivalent to an interrupt pending register whose source is TCC value, and CIER is similar to an interrupt enable register. Note that if the CIER*n* bit is disabled, the channel completion event is still registered in CIPR*n* if the channel parameter set is programmed with TCINT = 1 (or ATCINT = 1) and TCC = *n*. This is typical for a TCC/ATCC value used for chaining and not interrupt generation. If the CIER bit is subsequently enabled (before clearing CIPR), the EDMA_INT interrupt is sent to the CPU. If the CPU interrupt (defaults to CPU_INT8) is enabled, its interrupt service routine is executed.

In summary, to configure the EDMA for any channel (or QDMA request) to interrupt the CPU:

- Set CIE*n* to 1, in CIER
- Set TCINT (or ATCINT) to 1, in OPT
- Set TCC (or ATCC) to *n*, in OPT

1.15.1 EDMAINT Servicing by the CPU

When a completion code is detected by the EDMACC, the EDMACC sets the appropriate bit in CIPR as per the transfer complete code specified. The CPU ISR should read CIPR and determine what, if any, events/channels have completed and perform the necessary operations. The ISR should clear the bit in CIPR upon servicing the interrupt; therefore, enabling recognition of further interrupts. Writing a 1 to the relevant bit clears the CIPR bits, writing a 0 has no effect.

By the time one interrupt is serviced, many others could have occurred causing additional bits to be set in CIPR. Each of these bits in CIPR are likely to need different types of service by the CPU. The ISR should check for all pending interrupts and continue until all the posted interrupts are serviced.

After any write to CIPR, if the bitwise AND of CIPR and CIER is nonzero, the interrupt flag is set in the interrupt flag register (IFR) of the CPU. This implementation prevents losing interrupts that occur as the ISR is exited but can cause the ISR to be entered more than once. The additional call to the ISR occurs because the ISR is typically written to process and clear each CIPR bit serially. It is the write that clears the processed CIPR bit and that sets the additional IFR. The second time the ISR is called, the CIPR bit may be cleared to 0. As stated previously, the ISR should read CIPR and determine what, if any, events/channels have completed and perform the necessary operations. The second time the ISR is entered, if the CIPR is read as 0, no operations are necessary. To completely avoid the extra interrupt, clear all processed CIPR bits at once at the end of the ISR.

1.16 Quick DMA (QDMA)

Quick DMA (QDMA) provides an alternate means for the CPU to directly submit transfer requests (TR) to the EDMA transfer controller without using the EDMA channel controller. Since the QDMA registers are local to the CPU, the total amount of time for setting up a QDMA transfer is less than the time to set up a transfer in the EDMA channel controller. QDMA supports a subset of the transfer modes supported by the EDMA channel controller. In a typical system, the EDMA channel controller is used for periodic real-time peripheral servicing, such as providing the McBSP with transmit data at a regular rate. The QDMA is more appropriate for data that must be moved in blocks under direct control of the code running on the CPU.

1.16.1 Initiating a QDMA Transfer

A QDMA transfer requires only one to five CPU cycles to submit, depending on the number of registers that need to be configured. A typical QDMA transfer is performed by writing four of the parameter values to their registers followed by the write of the fifth parameter to its corresponding pseudo-register that initiates the transfer. All QDMA transfers are submitted to the EDMA transfer controller as a single transfer request that limits the QDMA functionality to a subset of the EDMA channel controller functionality. The QDMA supports frame-synchronized (1D) transfers of one frame (FRMCNT = 0) or block-synchronized (2D); therefore, the QDMA always requests a transfer of one complete frame (1D) or block (2D) of data. The value in the FS field of the QDMA channel options register (QOPT) is “don’t care”. There are no intermediate transfers in a QDMA transfer. Only one request is sent for any QDMA submission and Table 1−14 shows the number of elements transferred.

Table 1−14. QDMA Transfer Length

Transfer Dimension	Elements Transferred
1D to 1D	One frame, regardless of frame count
Other	One block, all arrays transferred

Thus, a typical submission sequence might look like:

```
QDMA_SRC = SOME_SRC_ADDRESS;
QDMA_DST = SOME_DST_ADDRESS;
QDMA_CNT = (NUMFRAME-1)<<16 | NUM_ELEMENTS; // Array Frame Count
QDMA_IDX = 0x00000000;            // no indexing specified
QDMA_S_OPT = 0x21B80001;          // frame synchronized 1D-SRC to 2D-DST,send
                                  // completion code 8 when finished
                                  // and submit transfer
```

All of the QDMA registers retain their value after the request is submitted, so if a second transfer is performed with any of the same parameter settings, they do not need to be rewritten by the CPU. Only the changed registers require rewriting, with the final parameter written to the appropriate pseudo-register to submit the transfer. As a result, subsequent QDMA requests can be submitted in as little as one CPU cycle per request.

1.16.2 QDMA Stalls and Priority

The QDMA has several stalling conditions. Once a write has been performed to one of the pseudo-registers (resulting in a pending QDMA transfer request), future writes to the QDMA registers stall until the transfer request is sent. Normally this will occur for 2−3 EDMA cycles, as this is how long it takes to submit a transfer. The CPU does not generally see stalls, because writes to the QDMA registers occur via the L1D write buffer. Future writes to the buffer may eventually fill it up and stall the CPU from subsequent reads/writes.

Similar to the EDMA channels, QDMA transfer controller priority is programmable, as described in section 3.4 and section 4.5. The PRI field in the QDMA channel options register (QOPT) specifies the TC priority level of the QDMA. On the C621x/C671x DSP, level 0 (urgent priority) is reserved for L2 cache accesses; thus, QDMA requests should only be submitted on a legal PRI level.

Refer to section 3.4.1.3 and section 4.5.1.3 for QDMA allocation issues with the QDMA.

1.17 Emulation Operation

During debug using the emulator, the CPU may be halted on an execute packet boundary for single stepping, benchmarking, profiling, or other debug uses. During an emulation halt, EDMA channel controller operations continue. Events continue to be captured/processed and transfers continue to be serviced.

1.18 Transfer Examples

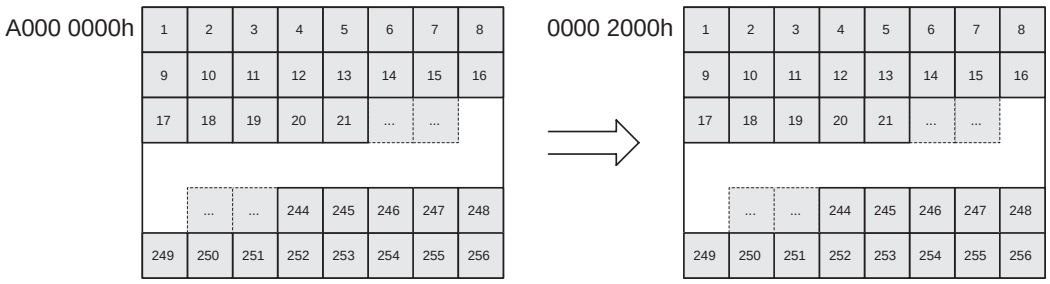
The EDMA channel controller performs a wide variety of transfers depending on the parameter configuration. The more basic transfers are performed either by an EDMA channel or by submitting a QDMA. More complicated transfers or repetitive transfers require the use of an EDMA channel. For a representation of various types of EDMA transfers, see Appendix A.

1.18.1 Block Move Example

The most basic transfer performed by the EDMA is a block move. During device operation it is often necessary to transfer a block of data from one location to another, usually between on-chip and off-chip memory.

In this example, a section of data is to be copied from external memory to internal L2 SRAM. The data block is 256 words and resides at address A000 0000h (CE2). The data is to be transferred to internal address 0000 2000h (L2 block 0), as shown in Figure 1−12.

Figure 1−12. Block Move Example Diagram



The fastest way to perform this transfer is through a QDMA request. The QDMA request can be submitted in several different ways, the most basic being a frame-synchronized 1D-to-1D transfer. This type of transfer is valid for block sizes of less than 64K elements. The transfer must be frame-synchronized so that all of the elements are transferred upon entry submission. QDMA submits all requests as frame-synchronized transfers, regardless of the FS bit value.

Figure 1−13 shows the parameters for this transfer. QDMA channel options, source address, destination address, and element count must be configured.

Figure 1−13. Block Move Example QDMA Registers Content

(a) QDMA Registers

Register Contents		Register
4120 0001h		QDMA Channel Options Register (QOPT)
A000 0000h		QDMA Channel Source Address Register (QSRC)
0000h	0100h	QDMA Channel Transfer Count Register (QCNT)
0000 2000h		QDMA Channel Destination Address Register (QDST)
Don't care	Don't care	QDMA Channel Index Register (QIDX)

(b) QDMA Channel Options Register (QOPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	01	0	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 1	0
0	00	0000 0000 0000	1
Rsvd	TCCM‡	Reserved	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

The source address for the QDMA is set to the start of the data block in external memory (A000 0000h), and the destination address is set to the start of the data block in L2 (0000 2000h). In QOPT all data is contiguous, so the SUM and DUM bits are both set to 01b (increment) and the PRI bit is set to low-priority for background transfer.

The CPU requires four cycles to submit the request for this transfer, one cycle for each register write. It requires fewer registers if any of the QDMA registers are already configured, with a minimum of one cycle. Three of the QDMA parameters must be written to their proper QDMA registers and one parameter must be written to its pseudo-register, initiating the transfer. A sample QDMA submission for the Figure 1−13 transfer follows:

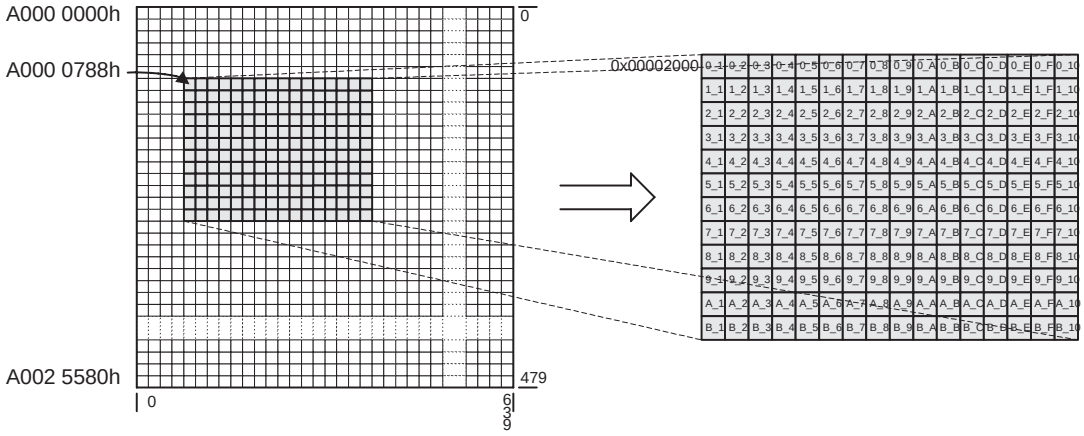
```
...
QDMA_SRC = 0xA0000000;   /* Set source address      */
QDMA_DST = 0x00002000;   /* Set destination address */
QDMA_CNT = 0x00000100;   /* Set frame/element count */
QDMA_S_OPT = 0x41200001; /* Set options and submit  */
...
```

A block that contains greater than 64K elements requires the use of both element count and array/frame count. Since the element count field is only 16 bits, the largest count value that can be represented is 65535. Any count larger than 65535 needs to be represented with an array count as well. A QDMA can still be used to transmit this amount of data. Rather than a frame-synchronized 1D-to-1D transfer, the QDMA needs to be configured as a block-synchronized (FS = 1) 2D-to-2D transfer.

1.18.2 Subframe Extraction Example

The EDMA can efficiently extract a small frame of data from a larger frame of data. By performing a 2D-to-1D transfer, the EDMA retrieves a portion of data for the CPU to process. In this example, a 640 × 480-pixel frame of video data is stored in external memory, CE2. Each pixel is represented by a 16-bit half-word. The CPU extracts a 16 × 12-pixel subframe of the image for processing. To facilitate more efficient processing time by the CPU, the EDMA places the subframe in internal L2 SRAM. Figure 1−14 depicts the transfer of the sub-frame from external memory to L2.

Figure 1−14. Subframe Extraction Example Diagram



To perform this transfer, the CPU can issue a QDMA request for a block-synchronized (FS = 1) 2D-to-1D transfer. Since the source is 2D and the transfer is block-synchronized, the QDMA requests a transfer of the entire subframe.

Figure 1−15 shows the parameters for this transfer. QDMA channel options, source address, destination address, and element count must be configured.

Figure 1−15. Subframe Extraction Example QDMA Registers Content

(a) QDMA Registers

Register Contents		Register
4D20 0001h		QDMA Channel Options Register (QOPT)
A000 0788h		QDMA Channel Source Address Register (QSRC)
000Bh	0010h	QDMA Channel Transfer Count Register (QCNT)
0000 2000h		QDMA Channel Destination Address Register (QDST)
04E0h	Don't care	QDMA Channel Index Register (QIDX)

(b) QDMA Channel Options Register (QOPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
010	01	1	01	0	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 1	0
0	00	0000 0000 0000	1
Rsvd	TCCM‡	Reserved	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

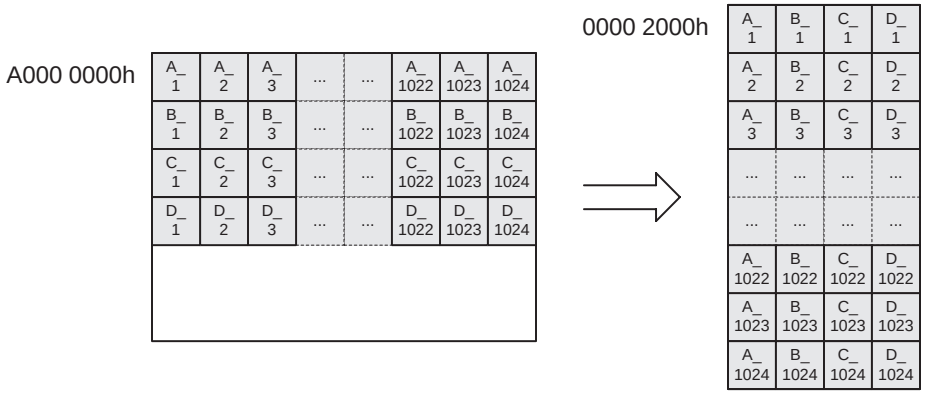
All of the address updates occur within the address generation/transfer logic. Therefore, the array index provided is the space between arrays of the subframe. Since each array of the video image is 640 pixels in length and each array of the subframe is 16 pixels in length, the array index is set to 2 bytes/element × (640 – 16) elements = 1248 bytes. The subframe is transferred to a block of contiguous memory. The element count (ELECNT) is set to 16, the number of elements per subframe array; the array count (FRMCNT) is set to 11, one less than the number of arrays. The QDMA request is sent to the low-priority queue so that it does not interfere with any potential data acquisition.

Inversely, a 1D-to-2D transfer can be used to perform the insertion of a subframe into a larger frame of data. In this example, the subframe could be inserted back into the larger image after some processing by the CPU.

1.18.3 Data Sorting Example

Many applications require the use of multiple data arrays; it is often desirable to have the arrays arranged such that the first elements of each array are adjacent, the second elements are adjacent, and so on. Often this is not how the data is presented to the device. Either data is transferred via a peripheral with the data arrays arriving one after the other or the arrays are located in memory with each array occupying a portion (frame) of contiguous memory spaces. For these instances, the EDMA can reorganize the data into the desired format. Figure 1−16 shows the data sorting of element arrays.

Figure 1−16. Data Sorting Example Diagram



Use the following values to determine the fields required to use QDMA requests to organize the data in memory by ordinal position:

- ❑ F = the initial value of frame count (FRMCNT)
- ❑ E = the initial value of element count (ELECNT), as well as the element count reload (ELERLD) value
- ❑ S = the element size in bytes

The QDMA transfers this data; however, due to the arrangement of the data in the destination, a single submission does not suffice. Instead, a separate QDMA transfer request is submitted for each frame. If an EDMA channel is necessary to perform this transfer, then an entry must be provided for each frame in the transfer in PaRAM. Also, the transfer must use the chaining feature to self-synchronize each frame on the completion of the previous frame.

This example shows equal sized data arrays that are located in external memory. The arrays do not need to be of equal length. If the lengths vary, each QDMA submission or each EDMA reload parameter set in PaRAM would contain the corresponding new count value.

For this example, assume that the 16-bit data is located in external RAM, beginning at address A000 0000h (CE2). The QDMA brings 4 frames of 1K halfwords from their locations in RAM to internal data memory beginning at 0000 2000h. The index value required is ELEIDX = F × S = 4 × 2 = 8.

Since separate QDMA transfer requests are submitted for each frame, the QDMA parameters only use ELEIDX. The CPU updates the destination address for each new frame. For the first frame of data, the values shown in Figure 1−17 are assigned to the QDMA registers. For each subsequent frame, the CPU must perform two stores to change the source address and the destination address. The CPU does not need to wait for each frame to complete before submitting a request for the next. The transfer queues store the subsequent transfer requests to await processing.

Figure 1−17. Data Sorting Example QDMA Registers Content

(a) QDMA Registers

Register Contents		Register
4960 0001h		QDMA Channel Options Register (QOPT)
A000 0000h		QDMA Channel Source Address Register (QSRC)
0000h	0400h	QDMA Channel Transfer Count Register (QCNT)
0000 2000h		QDMA Channel Destination Address Register (QDST)
Don't care	0008h	QDMA Channel Index Register (QIDX)

(b) QDMA Channel Options Register (QOPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
010	01	0	01	0	11	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 1	0
0	00	0000 0000 0000	1
Rsvd	TCCM‡	Reserved	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

To summarize, the CPU performs four writes to configure the channel options, the source address, the count, and the destination address. The CPU then performs a write to the channel index pseudo-register (or the register is still not configured) to submit the transfer request for the first frame. For each additional frame, the CPU increments the source address by $E \times S = 1024 \times 2 = 2048$ and stores this value to the source address register (QSRC), and also increments the destination address by S and stores this value to the destination address pseudo-register (QSDST) to submit the transfer request.

If it is desired to have the EDMA notify the CPU when all of the transfers have completed, then the transfer request for the last frame should also have a modified channel options field to include a transfer complete code value and set TCINT = 1.

1.18.4 Peripheral Servicing Examples

The EDMA channel controller also services peripherals in the background of CPU operation, without requiring any CPU intervention. Through proper initialization of the EDMA channels, they can be configured to continuously service on-chip and off-chip peripherals throughout the device operation. Each event available to the EDMA has its own dedicated channel, and all channels operate simultaneously.

Since all EDMA channels are always synchronized, there are no special setups required to configure a channel to properly service a particular event. The only requirements are to use the proper channel for a particular transfer and to enable the channel event in the event enable register (EER) or channel chain enable register (CCER), unless the CPU synchronizes the channel.

When programming an EDMA channel to service a peripheral, it is necessary to know how data is to be presented to the DSP. Data is always provided with some kind of synchronization event as either one element per event (non-bursting) or multiple elements per event (bursting).

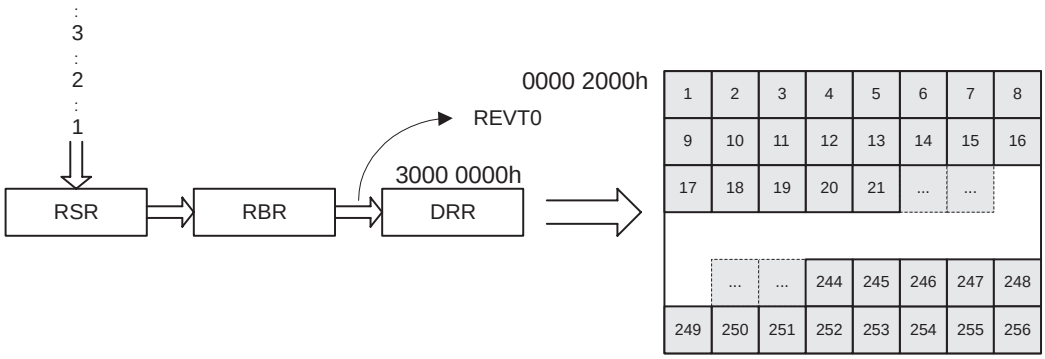
1.18.4.1 Non-bursting Peripherals

Non-bursting peripherals include the on-chip multichannel buffered serial port (McBSP) and many external devices, such as codecs. Regardless of the peripheral, the EDMA channel configuration is the same.

The on-chip McBSP is the most commonly-used peripheral in a C6000 DSP system. EDMA channels 12 and 13 are mapped to McBSP0 transmit and receive events, and channels 14 and 15 are mapped to McBSP1 transmit and receive events. The transmit and receive data streams are treated independently by the EDMA. While a standard DMA channel could be used in split-mode to handle transmit and receive data, there are a number of restrictions with this due to the sharing of resources. The EDMA channels do not have these restrictions. Although most serial applications call for similar data formats both to and from the McBSP, this is not a requirement for reliable operation with the EDMA. The transmit and receive data streams can have completely different counts, data sizes, and formats.

If the block move example, section 1.18.1, were changed such that the 256 words were received by McBSP0 to be transferred to internal L2 SRAM, this is easily handled by EDMA channel 13, which is synchronized by REVT0. Figure 1−18 shows a diagram of this transfer.

Figure 1−18. Servicing Incoming McBSP Data Example Diagram



To transfer the incoming data stream to its proper location in L2 memory, the EDMA channel must be set up for a 1D-to-1D transfer with element synchronization (FS = 0). Since an event (REVT0) is generated for every word as it arrives, it is necessary to have the EDMA issue the transfer request for each element individually. Figure 1−19 shows the parameters for this transfer.

The source address of the EDMA channel is set to the data receive register (DRR) address for McBSP0, and the destination address is set to the start of the data block in L2. Since the address of DRR is fixed, the SUM bit is cleared to 00b (no modification) and the DUM bit is set to 01b (increment). The priority level (PRI) chosen in this example is based on the premise that serial data is typically a high priority, so that samples are not missed. Each transfer request by this channel is made on the high-priority queue (Q1).

Figure 1−19. Servicing Incoming McBSP Data Example EDMA Parameters Content

(a) EDMA Parameters

Parameter Contents		Parameter
2020 0000h		EDMA Channel Options Parameter (OPT)
3000 0000h		EDMA Channel Source Address (SRC)
0000h	0100h	EDMA Channel Transfer Count (CNT)
0000 2000h		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(b) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	00	0	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

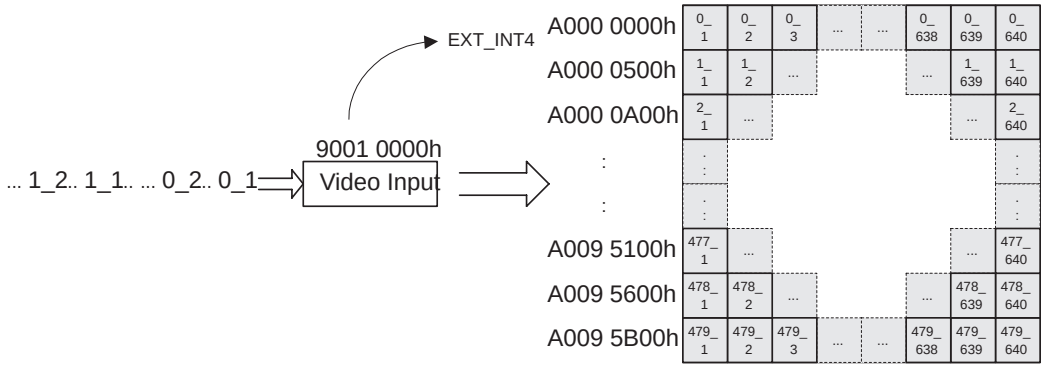
† TCCM is reserved on C621x/C671x DSP.

1.18.4.2 Bursting Peripherals

Higher bandwidth applications require that multiple data elements be presented to the DSP for every sync event. This frame of data can either be from multiple sources that are working simultaneously or from a single high-throughput peripheral that streams data to/from the DSP.

In this example, a video framer is receiving a video frame from a camera and presenting it to the DSP one array at a time. The video image is 640 × 480 pixels, with each pixel represented by a 16-bit element. The image is to be stored in external memory. Figure 1−20 shows a diagram of this transfer.

Figure 1−20. Servicing Peripheral Bursts Example Diagram



To transfer data from an external peripheral to an external buffer one array at a time based on EXT_INT4, channel 4 must be configured. There are two types of transfers that are suitable and are functionally identical: a 1D-to-1D transfer with frame synchronization (FS = 1) or a 1D-to-2D transfer with array synchronization (FS = 0). Due to the nature of the data (a video frame made up of arrays of pixels) the destination is essentially a 2D entity. Figure 1−21 shows the parameters to service the incoming data with a 1D-to-2D transfer.

The source address is set to the location of the video framer peripheral, and the destination address to the start of the data buffer. Since the input address is static, the SUM bit is cleared to 00b (no modification). The destination is made up of arrays of contiguous, linear elements; therefore, the DUM bit is set to 01b (increment). The element count (ELECNT) is equal to the number of pixels in an array, 640. The array count (FRMCNT) is equal to 1 less than the total number of arrays in the block, 479. An array index that is equal to the difference between the starting addresses of each array is required. Since each pixel is represented by a halfword, the array index (FRMIDX) is equal to twice the element count, 640 × 2 = 1280 bytes.

Figure 1−21. Servicing Peripheral Bursts Example EDMA Parameters Content

(a) EDMA Parameters

Parameter Contents		Parameter
28A0 0000h		EDMA Channel Options Parameter (OPT)
9001 0000h		EDMA Channel Source Address (SRC)
01DFh	0280h	EDMA Channel Transfer Count (CNT)
A000 0000h		EDMA Channel Destination Address (DST)
0500h	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(b) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
001	01	0	00	1	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

1.18.4.3 Continuous Operation

Configuring an EDMA channel to receive a single frame of data can be useful, and is applicable to some systems. A majority of the time, however, data is going to be continuously transmitted and received throughout the entire operation of the DSP. In this case, it is necessary to implement some form of linking such that the EDMA channels continuously reload the necessary parameter sets.

In this example, McBSP0 is configured to transmit and receive data on a T1 array. To simplify the example, only two channels are active for both transmit and receive data streams. Each channel receives packets of 128 elements. The packets are transferred from the serial port to L2 memory and from L2 memory to the serial port, as shown in Figure 1−22.

The McBSP generates REVT0 for every element received and generates XEVT0 for every element transmitted. To service the data streams, EDMA channels 12 and 13 must be set up for 1D-to-1D transfers with element synchronization (FS = 0). Figure 1−23 shows the parameters for the channel entries for these transfers.

In order to service the McBSP continuously throughout DSP operation, the channels must be linked to a duplicate entry in the PaRAM. After all frames have been transferred, the EDMA channels reload and continue. Figure 1−24 shows the reload parameters for the channel entries for these transfers.

Figure 1−22. Servicing Continuous McBSP Data Example Diagram

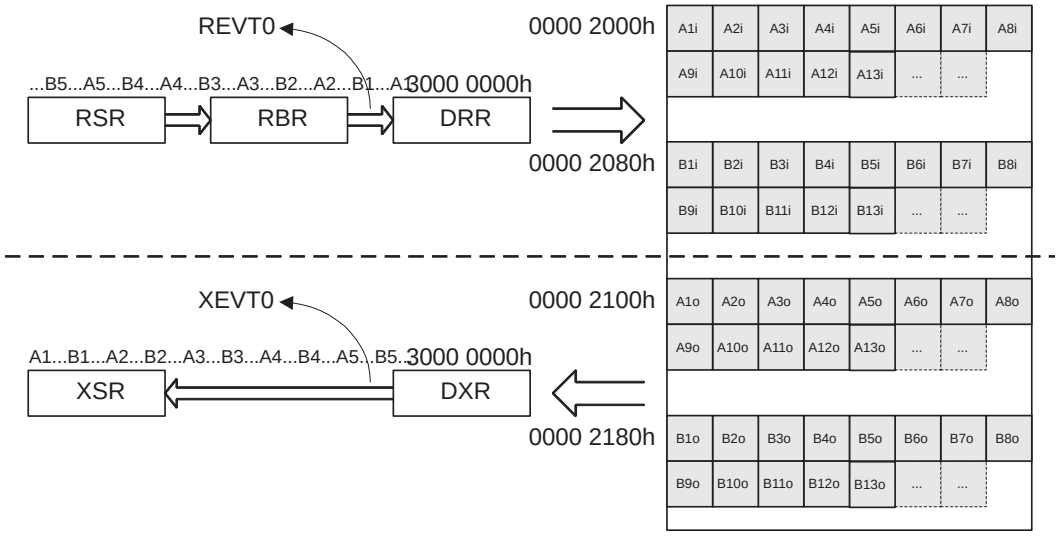


Figure 1−23. Servicing Continuous McBSP Data Example EDMA Parameters Content

(a) EDMA Parameters for Receive Channel 13

Parameter Contents		Parameter
3060 0002h		EDMA Channel Options Parameter (OPT)
3000 0000h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
0000 2000h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	0198h	EDMA Channel Count Reload/Link Address (RLD)

(b) EDMA Channel Options Parameter (OPT) Content for Receive Channel 13

31 29	28 27	26	25 24	23	22 21	20	19 16
001	10	0	00	0	11	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	1	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

(c) EDMA Parameters for Transmit Channel 12

Parameter Contents		Parameter
3300 0002h		EDMA Channel Options Parameter (OPT)
0000 2100h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
3000 0000h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	0180h	EDMA Channel Count Reload/Link Address (RLD)

(d) EDMA Channel Options Parameter (OPT) Content for Transmit Channel 12

31 29	28 27	26	25 24	23	22 21	20	19 16
001	10	0	11	0	00	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	1	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure 1−24. Servicing Continuous McBSP Data Example EDMA Reload Parameters Content

(a) EDMA Reload Parameters for Receive Channel 13 (Address 01A0 0198h)

Parameter Contents		Parameter
3060 0002h		EDMA Channel Options Parameter (OPT)
3000 0000h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
0000 2000h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	0198h	EDMA Channel Count Reload/Link Address (RLD)

(b) EDMA Reload Parameters for Transmit Channel 12 (Address 01A0 0180h)

Parameter Contents		Parameter
3300 0002h		EDMA Channel Options Parameter (OPT)
0000 2100h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
3000 0000h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	0180h	EDMA Channel Count Reload/Link Address (RLD)

Receive Channel

EDMA channel 13 services the incoming data stream of McBSP0. The source address is set to that of the data receiver register (DRR), and the destination address is set to the first element of the data block. Since there are two data channels being serviced, A and B, and they are to be located separately within the L2 SRAM, the destination address update mode uses element and frame indexing (DUM = 11b). The element index is the offset between the first element of each channel's data section. The frame index is the offset between the second element of channel A and the first element of channel B. Since elements are 8-bit, the ESIZE field is set to 10b.

In order to facilitate continuous operation, a copy of the channel entry is placed in parameter RAM at address 01A0 1980h. The LINK option is set and the link address is provided in the parameter entry. Upon exhausting the channel 13 element and frame counts, the parameters located at the link address are loaded into the channel 13 parameter set and operation continues. This function continues throughout DSP operation until halted by the CPU.

The parameter table must keep track of the element count within the frame since each element is sent individually (FS = 0). It is required that an element count reload be provided in the parameter set. This value is reloaded to the element count field every time the element count reaches 0.

Transmit Channel

EDMA channel 12 services the outgoing data stream of McBSP0. Its configuration is essentially the opposite of channel 13 for this application since the input and output data is symmetrical. The element and frame counts, and the index values are identical. The channel options are reversed, such that the source address is updated using the programmed index values while the destination address is held constant. The source address provided to the channel is that of the beginning of channel A output data, and the destination address is that of the data transmit register (DXR). Linking is also used to allow for continuous operation by the EDMA channel, with a duplicate entry in the parameter RAM.

1.18.4.4 Ping-Pong Buffering

Although the previous configuration allows the EDMA to service a peripheral continuously, it presents a number of restrictions to the CPU. Since the input and output buffers are continuously being filled/emptied, the CPU must match the pace of the EDMA very closely in order to process the data. The EDMA receive data must always be placed in memory before the CPU accesses it, and the CPU must provide the output data before the EDMA transfers it. Though not impossible, this is an unnecessary challenge. It is particularly difficult in a two-level cache system.

Ping-pong buffering is a simple technique that allows the CPU activity to be distanced from the EDMA activity. This means that there are multiple (usually two) sets of data buffers for all incoming and outgoing data streams. While the EDMA transfers the data into and out of the ping buffers, the CPU manipulates the data in the pong buffers. When both CPU and EDMA activity completes, they switch. The EDMA then writes over the old input data and transfers the new output data. Figure 1−25 shows the ping-pong scheme for this example.

To change the continuous operation example, section 1.18.4.3, such that a ping-pong buffering scheme is used, the EDMA channels need only a moderate change. Instead of one parameter set, there are two; one for transferring data to/from the ping buffers and one for transferring data to/from the pong buffers. As soon as one transfer completes, the channel loads the entry for the other and the data transfers continue. Figure 1−26 shows the EDMA channel configuration required.

Each channel has two parameter sets, ping and pong. The EDMA channel is initially loaded with the ping parameters (Figure 1−26). The link address for the ping entry is set to the PaRAM offset of the pong parameter set (Figure 1−27). The link address for the pong entry is set to the PaRAM offset of the ping parameter set (Figure 1−28). The channel options, count values, index values are all identical between the ping and pong parameters for each channel. The only differences are the link address provided and the address of the data buffer in internal memory.

Synchronization with the CPU

In order to utilize the ping-pong buffering technique, the system must signal the CPU when to begin to access the new data set. After the CPU finishes processing an input buffer (ping), it waits for the EDMA to complete before switching to the alternate (pong) buffer.

In this example, both channels provide their channel numbers as their report word and set the TCINT bit to 1 to generate an interrupt after completion. When channel 13 fills an input buffer, the CIP13 bit is set to 1 in the channel interrupt pending register (CIPR); when channel 12 empties an output buffer, the CIP12 bit is set to 1 in CIPR. The CPU must manually clear these bits.

With the channel parameters set, the CPU polls CIPR to determine when to switch. The EDMA and CPU could alternatively be configured such that the channel completion interrupts the CPU. By doing this, the CPU could service a background task while waiting for the EDMA to complete.

Figure 1−25. Ping-Pong Buffering for McBSP Data Example Diagram

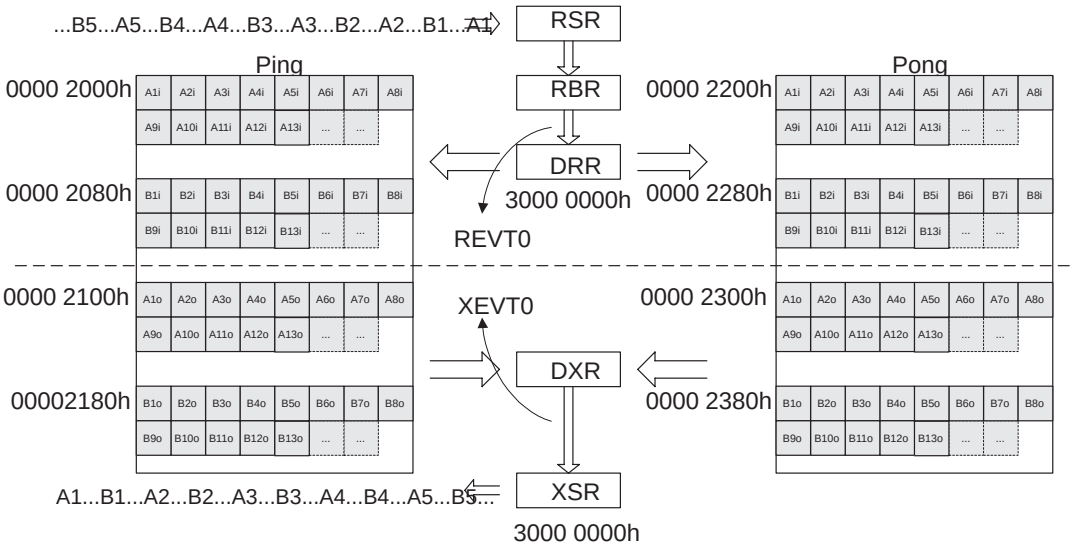


Figure 1−26. Ping-Pong Buffering Example EDMA Parameters Content

(a) EDMA Parameters for Channel 13

Parameter Contents		Parameter
307D 0002h		EDMA Channel Options Parameter (OPT)
3000 0000h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
0000 2000h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	01B0h	EDMA Channel Count Reload/Link Address (RLD)

(b) EDMA Channel Options Parameter (OPT) Content for Channel 13

31 29	28 27	26	25 24	23	22 21	20	19 16
001	10	0	00	0	11	1	1101
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	1	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

(c) EDMA Parameters for Channel 12

Parameter Contents		Parameter
331C 0002h		EDMA Channel Options Parameter (OPT)
0000 2100h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
3000 0000h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	0180h	EDMA Channel Count Reload/Link Address (RLD)

(d) EDMA Channel Options Parameter (OPT) Content for Channel 12

31 29	28 27	26	25 24	23	22 21	20	19 16
001	10	0	11	0	00	1	1100
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	1	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure 1−27. Ping-Pong Buffering Example EDMA Pong Parameters Content

(a) EDMA Pong Parameters for Channel 13 at Address 01A0 01B0h

Parameter Contents		Parameter
307D 0002h		EDMA Channel Options Parameter (OPT)
3000 0000h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
0000 2200h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	01C8h	EDMA Channel Count Reload/Link Address (RLD)

(b) EDMA Pong Parameters for Channel 12 at Address 01A0 0180h

Parameter Contents		Parameter
331C 0002h		EDMA Channel Options Parameter (OPT)
0000 2300h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
3000 0000h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	0198h	EDMA Channel Count Reload/Link Address (RLD)

Figure 1−28. Ping-Pong Buffering Example EDMA Ping Parameters Content

(a) EDMA Ping Parameters for Channel 13 at Address 01A0 01C8h

Parameter Contents		Parameter
307D 0002h		EDMA Channel Options Parameter (OPT)
3000 0000h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
0000 2000h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	01B0h	EDMA Channel Count Reload/Link Address (RLD)

(b) EDMA Ping Parameters for Channel 12 at Address 01A0 0198h

Parameter Contents		Parameter
331C 0002h		EDMA Channel Options Parameter (OPT)
0000 2100h		EDMA Channel Source Address (SRC)
007Fh	0002h	EDMA Channel Transfer Count (CNT)
3000 0000h		EDMA Channel Destination Address (DST)
FF81h	0080h	EDMA Channel Index (IDX)
0002h	0180h	EDMA Channel Count Reload/Link Address (RLD)

1.18.5 Transfer Chaining Examples

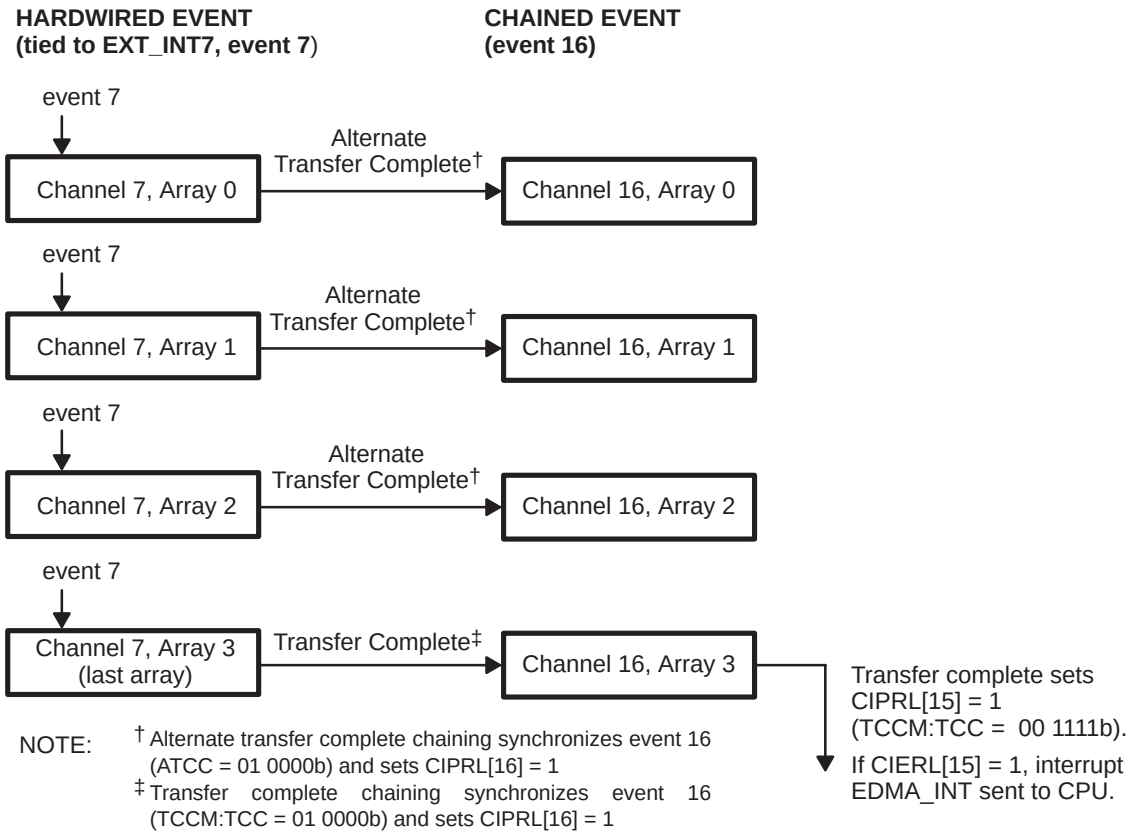
The following examples explain the alternate transfer complete chaining function in detail.

1.18.5.1 Servicing Input/Output FIFOs with a Single Event

Most common systems for ADSL, networking, and video applications require the use of a pair of external FIFOs that must be serviced at the same rate. One FIFO buffers data input, and the other buffers data output. The EDMA channels that service these FIFOs can be set up for 2D array synchronized (FS = 0) transfers. While each FIFO is serviced with a different set of parameters, both can be signaled from a single event. For example, an external interrupt pin can be tied to the status flags of one of the FIFOs. When this event arrives, the EDMA needs to perform servicing for both the input and output streams. Without the alternate transfer complete chaining feature this would require two events, and thus two external interrupt pins. The alternate transfer complete chaining feature allows the use of a single external interrupt pin (for example, EXT_INT7). Figure 1−29 shows the EDMA setup and illustration for this example.

An EXT_INT7 event triggers a channel 7 array transfer. Upon completion of each intermediate array transfer of channel 7, alternate transfer complete chaining sets bit CIPRL[16] (specified by channel 7 ATCC) and provides a synchronization event to channel 16. Upon completion of the last array transfer of channel 7, transfer complete chaining—not alternate transfer complete chaining—sets bit CIPRL[16] (specified by its TCCM:TCC) and provides a synchronization event to channel 16. The completion of channel 16 sets bit CIPRL[15] (specified by its TCCM:TCC), which can generate an interrupt to the CPU, if CIERL[15] = 1.

Figure 1−29. Alternate Transfer Complete Chaining Example



SETUP

Channel 7 Parameters for Chaining

- ❑ Enable Transfer Complete Chaining:
  TCINT = 1
  TCCM:TCC = 01 0000b
- ❑ Enable Alternate Transfer Complete Chaining
  ATCINT = 1
  ATCC = 01 0000b

Channel 16 Parameters for Chaining

- ❑ Enable Transfer Complete Chaining:
  TCINT = 1
  TCCM:TCC = 00 1111b
- ❑ Disable Alternate Transfer Complete Chaining
  ATCINT = 0
  ATCC = don't care

Event Enable Register (EER)

- ❑ Enable channel 7:
  EERL[7] = 1

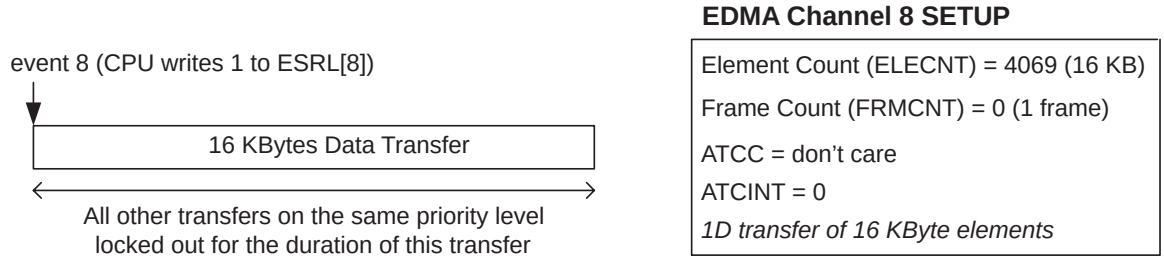
Channel Chain Enable Register (CCER)

- ❑ Enable chaining to channel 16: CCERL[16] = 1

1.18.5.2 Breaking up Large Transfers with ATCC

Another feature of the alternate transfer completion code (ATCC) is for breaking up large transfers. A large transfer may lock out other transfers of the same priority level (section 4.5) for the duration of the transfer. For example, a large transfer with high priority from the internal memory to the external memory using the EMIF may lock out other EDMA transfers on the high priority queue. In addition, this large high priority transfer may lock out the EMIF for a long period of time from lower priority channels. When a large transfer is considered to be high priority, it should be split into multiple smaller transfers. Figure 1−30 shows the EDMA setup and illustration of an example single large block transfer.

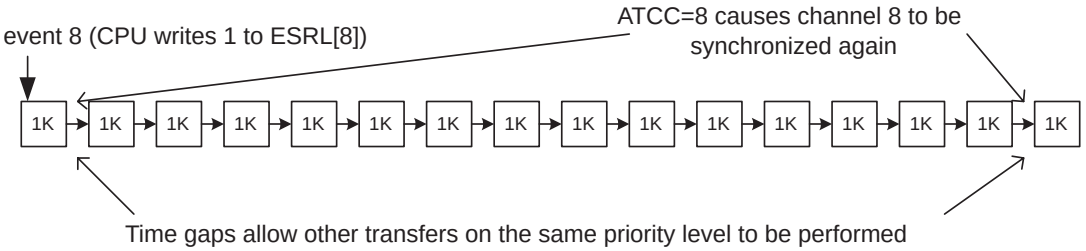
Figure 1−30. Single Large Block Data Transfer Example



The alternate transfer completion code (ATCC) provides a method to break up a large transfer into smaller transfers. For example, to move a single large block of memory (16K bytes), the EDMA will perform a 2D array synchronized transfer. The element count is set to a “reasonable” value, where reasonable derives from the amount of time it would take to move this smaller amount of data. Assume 1 Kbyte is a reasonable small transfer in this example. The EDMA is set up to transfer 16 arrays of 1 Kbyte elements, for a total of 16K byte elements. The ATCC field in the channel options parameter (OPT) is set to the same value as the channel number. In this example, EDMA channel 8 is used and ATCC is also set to 8. The transfer complete code TCCM:TCC may be set to a different value to cause an interrupt to the CPU at the end of the transfer.

The CPU starts the EDMA transfer by writing to the appropriate bit of the event set register (ESRL[8]). The EDMA transfers the first 1 Kbyte array. Upon completion of the first array (an intermediate transfer), alternate transfer complete code chaining generates a synchronization event to channel 8, a value specified by the ATCC field. This ATCC-generated synchronization event causes EDMA channel 8 to transfer the next 1 Kbyte array. This process continues until the transfer parameters are exhausted, at which point the EDMA has completed the 16 Kbyte transfer. This method breaks up a large transfer into smaller packets, thus providing natural time slices in the transfer such that other events may be processed. Figure 1−31 shows the EDMA set-up and illustration of the broken up smaller packet transfers.

Figure 1−31. Smaller Packet Data Transfers Example



EDMA Channel 8 SETUP

Element Count (ELECNT) = 256 (1KB)	ATCC = 8
Array Count (FRMCNT) = 15 (16 frames)	ATCINT = 1
Array Index (FRMIDX) = 1024 (1KB strides)	
Array sync (FS) = 0	
CCER[8] = 1†	*2D transfer of 16 1KByte arrays*

† If another channel that has a system event (such as channel 4) is used, both the event enable (EER[4]) and channel chain enable (CCER[4]) must be set for the channel to transfer after both the external sync and the ATCC are received.

Chapter 2

EDMA Transfer Controller

This chapter describes the EDMA transfer controller (EDMATC) that handles all data transfers between the level-two (L2) cache/memory controller and the device peripherals on the C621x/C671x/C64x DSP. These data transfers include EDMA channel controller transfers, cache accesses to/from EMIF range, noncacheable memory accesses, and master peripheral accesses.

Topic **Page**

2.1 EDMA Transfer Controller Performance

EDMA bandwidth is fully utilized when performing burst transfer, which is obtained if and only if the EDMA transfer is configured as follows:

- Transfer/synchronization type is array-/frame-/block-synchronized transfer (not element-synchronized, see section 1.9.1)
- Element size is 32-bit (ESIZE = 00b)
- Addressing mode is increment, decrement, or fixed (not indexed, SUM or DUM = 00/01/10b in the options parameter)

The EDMA performs single element transfers for all transfers not meeting the above conditions, which utilizes only a portion of the bandwidth available.

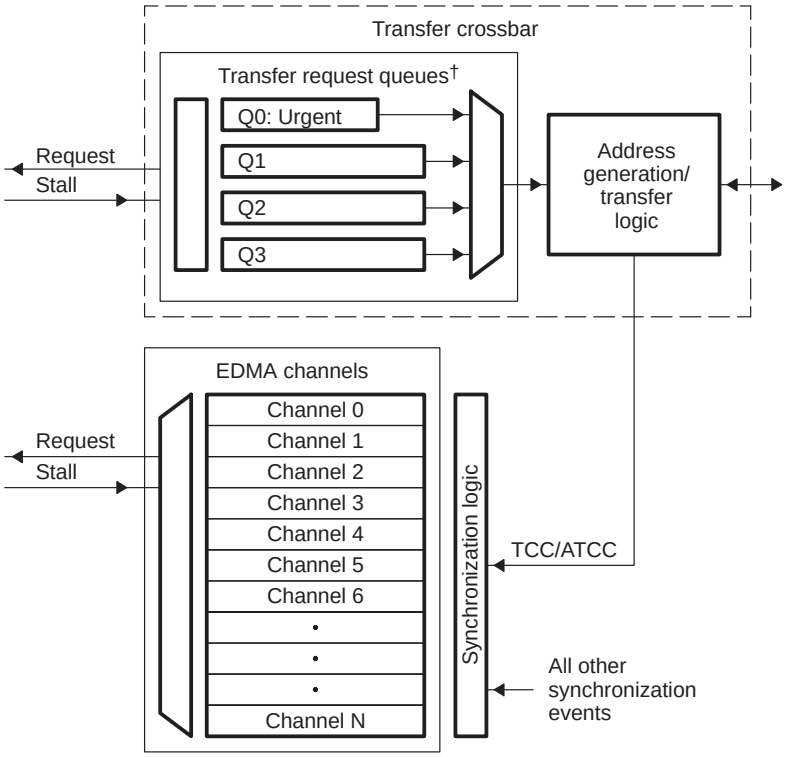
For the burst transfer types described above, the burst length is dictated by the 1D component of the transfer, which is specified by the ELECNT field. For array- or frame-synchronized transfer, the 1D component of the transfer is the amount of data that gets transferred per synchronization event. For block-synchronized transfers, the complete 2D transfer is transferred per synchronization event; however, burst transfers are only performed for the 1D component. If the 1D length (ELECNT) is programmed to a small value, the performance reduces accordingly and in the worst case (ELECNT = 1), the performance is identical to the performance described for single element transfers.

2.2 Transfer Request Submission

2.2.1 Request Chain

All transfer requestors to the EDMA are connected to the transfer request chain, as shown in Figure 2−1. A transfer request, once submitted, shifts through the chain to the transfer crossbar (TC), where it is prioritized and processed. The transfer request can be for a single data element or for a large number of elements, as described in sections 1.5 and 1.4.

Figure 2−1. EDMA Transfer Request Block Diagram



† Q3 is available to C64x DSP only.

2.2.1.1 L2 Controller Transfer Requests

The L2 controller submits all transfer requests for cache servicing, for accessing noncacheable memory, and for QDMA transfers. See the *Two-Level Internal Memory Reference Guides* (SPRU609 and SPRU610) for details on the cacheability of memory.

For C621x/C671x DSP, cache servicing requests are always made on the urgent priority level. For read requests, the cache controller always requests an L2 line in two parts, requesting the “missed” portion of the line first. The data transfers requested are based on the data location within the L2 line, as shown in Table 2−1. For write requests, as a result of writeback/writeback-invalidate operations or eviction, the burst size is one complete L2 line.

Table 2−1. Cache Controller Data Transfers—C621x/C671x DSP

Data Location	First Transfer	Second Transfer
First 1/4	Front 1/2 line	Back 1/2 line
Second 1/4	Back 3/4	Front 1/4 line
Third 1/4	Back 1/2 line	Front 1/2 line
Fourth 1/4	Back 1/4 line	Front 3/4 line

For C64x DSP, cache servicing requests can be made on any priority levels as specified in the P bits of the cache configuration register (CCFG). For read requests, the cache controller always requests an L2 line in two bursts of 64K bytes each, requesting the “missed” portion of the line first. For write requests, as a result of writeback/writeback-invalidate operations or eviction, the cache controller transfers one complete L2 line in two bursts of 64 bytes each.

For both C621x/C671x DSP and C64x DSP, transfer requests by the L2 controller for noncacheable memory are always equal to a single element and are used to load/store data from/to a noncacheable location in external memory.

QDMA transfer requests have the same restrictions as the EDMA channels. See section 1.16 for details.

2.2.1.2 HPI/PCI Transfer Requests

The HPI/PCI automatically generates transfer requests to service host activity. For C621x/C671x DSP, these transfer request submissions are submitted only with a high priority and are invisible. For C64x DSP, by default HPI/PCI transfer requests are submitted with medium priority, but request priority can be programmed to any of the four priority levels by setting the PRI field in the transfer request control register (TRCTL) to the appropriate value. The HPI/PCI submits a transfer request for a single element read or write for fixed mode host accesses and a transfer request for a short data burst for autoincrement transfers. The burst size is always for eight or fewer elements. See section 3.4 and section 4.5 for available HPI transfer request priority.

2.2.1.3 EDMA Channel Transfer Requests

The EDMA channel transfers can be submitted with urgent (C64x DSP only), high, medium (C64x DSP only), or low priority; with the recommendation that high priority be reserved for short bursts and single element transfers and low priority be used for longer (background) block moves. It is also recommended to divide transfers between the priority levels when applicable, as this helps to maximize the device performance. See section 3.4 and section 4.5 for available EDMA channel transfer request priority.

2.2.2 Transfer Crossbar

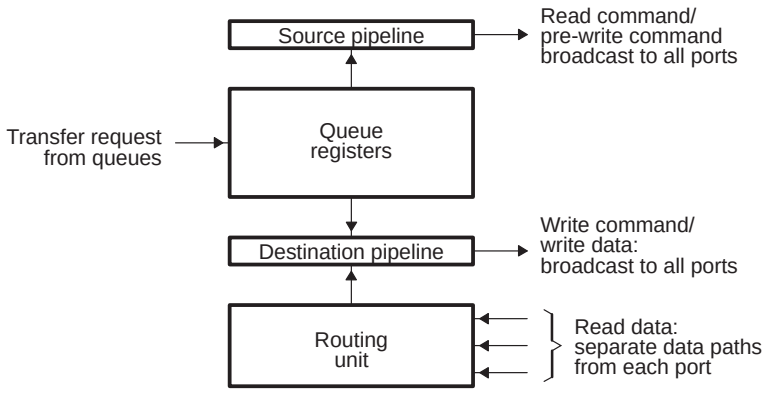
Once a transfer request reaches the end of the request chain, it is sent to the transfer crossbar (TC). Within TC, the transfer request shifts into one of the transfer request queues to await processing. The queue to which it is submitted is determined by its priority associated. The C621x/C671x DSP has three fixed-length queues (Q0−Q2), the C64x DSP has four priority level queues (Q0−Q3) with programmable lengths.

Once the transfer request reaches the head of its queue, it is submitted to the address generation/transfer logic for processing. The address generation/transfer logic only services one transfer request from each queue. To maximize the data transfer bandwidth in a system, utilize all queues.

2.2.3 Address Generation/Transfer Logic

The address generation/transfer logic block, shown in Figure 2−2, controls the transferring of data by the EDMA. Each priority queue has one register set, which monitors the progress of a transfer. Within the register set for a particular queue, the current source address, destination address, and count are maintained for a transfer. These registers are not memory-mapped and are not available to the CPU.

Figure 2−2. Address Generation/Transfer Logic Block Diagram



Chapter 3

TMS320C621x/C671x EDMA

This chapter describes the operation and registers of the EDMA controller in the TMS320C621x/C671x DSP. This chapter also describes the quick DMA (QDMA) registers that the CPU uses for fast data requests. For operation and registers unique to the TMS320C64x™ EDMA, see Chapter 4.

Topic **Page**

3.1 Event Service Priority

The EDMA event register (ER) captures up to 16 events; therefore, it is possible for events to occur simultaneously on the EDMA event inputs. For events arriving simultaneously, the channel with the highest event number submits its transfer request first. This mechanism only sorts simultaneous events and does nothing to set the priority of the events. The priority of the transfer is determined by its EDMA parameters stored in the parameter RAM (PaRAM) of the EDMA.

3.2 Parameter RAM (PaRAM)

Unlike the C620x/C670x DMA controller, which is a register-based architecture, the EDMA controller is a RAM-based architecture. EDMA channels are configured in a parameter table. The table is a 2-KByte block of parameter RAM (PaRAM) located within the EDMA channel controller. The PaRAM table (Table 3−1) consists of six-word parameter entries of 24 bytes each, for a total of 85 entries. The contents of the 2-KByte PaRAM include:

- 16 transfer parameter set entries for the 16 EDMA events. Each parameter set is six words or 24 bytes. These areas can also serve as reload/link parameters.
- Remaining parameter sets serve as additional parameters used for reloading or linking transfers. Each reload/link parameter set is 24 bytes.

Each parameter entry of an EDMA event is organized into six 32-bit words or 24 bytes, as shown in Figure 3−1 and described in Table 3−2.

Table 3−1. EDMA Parameter RAM Contents—C621x/C671x DSP

Address	Parameters
01A0 0000h to 01A0 0017h	Parameters for event 0 (6 words)
01A0 0018h to 01A0 002Fh	Parameters for event 1 (6 words)
01A0 0030h to 01A0 0047h	Parameters for event 2 (6 words)
01A0 0048h to 01A0 005Fh	Parameters for event 3 (6 words)
01A0 0060h to 01A0 0077h	Parameters for event 4 (6 words)
01A0 0078h to 01A0 008Fh	Parameters for event 5 (6 words)
01A0 0090h to 01A0 00A7h	Parameters for event 6 (6 words)
01A0 00A8h to 01A0 00BFh	Parameters for event 7 (6 words)
01A0 00C0h to 01A0 00D7h	Parameters for event 8 (6 words)
01A0 00D8h to 01A0 00EFh	Parameters for event 9 (6 words)
01A0 00F0h to 01A0 0107h	Parameters for event 10 (6 words)
01A0 0108h to 01A0 011Fh	Parameters for event 11 (6 words)
01A0 0120h to 01A0 0137h	Parameters for event 12 (6 words)
01A0 0138h to 01A0 014Fh	Parameters for event 13 (6 words)
01A0 0150h to 01A0 0167h	Parameters for event 14 (6 words)
01A0 0168h to 01A0 017Fh	Parameters for event 15 (6 words)
01A0 0180h to 01A0 0197h	Reload/link parameter for Event 0 (6 words)†
01A0 0198h to 01A0 01AFh	Reload/link parameter for Event 1 (6 words)†
...	...
01A0 07E0h to 01A0 07F7h	Reload/link parameter for Event 68 (6 words)†
01A0 07F8h to 01A0 07FFh	Scratch pad area (2 words)

† The C621x/C671x DSP has 69 parameter sets [6-words each] that can be used to reload/link EDMA transfers.

Figure 3−1. EDMA Channel Parameter Entries for Each EDMA Event—C621x/C671x DSP

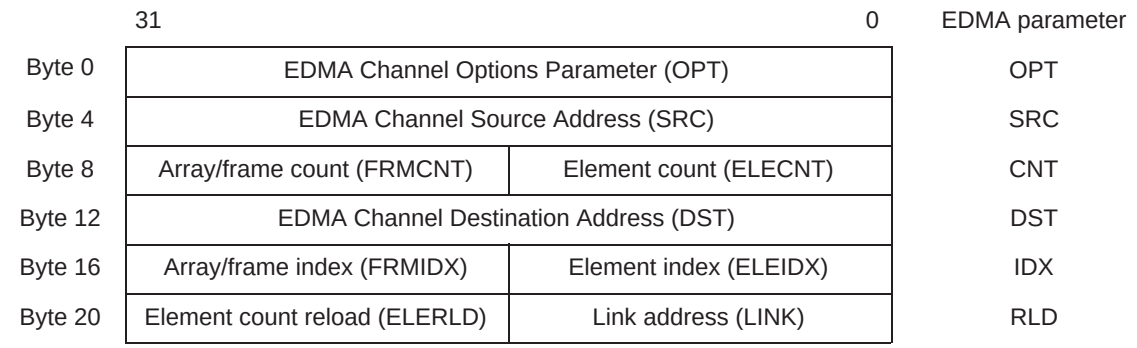


Table 3−2. EDMA Channel Parameter Descriptions—C621x/C671x DSP

Offset address (bytes)	Acronym	Parameter	As defined for… 1D transfer	As defined for… 2D transfer	Section
0	OPT	Channel options	Transfer configuration options.		3.6.1
4	SRC	Channel source address	The address from which data is transferred.		3.6.2
8†	ELECNT	Element count	The number of elements per frame.	The number of elements per array.	3.6.3
	FRMCNT	Frame count (1D), Array count (2D)	The number of frames per block minus 1.	The number of arrays per block minus 1.	3.6.3
12	DST	Channel destination address	The address to which data is transferred.		3.6.4
16†	ELEIDX	Element index	The address offset of elements within a frame.	−−−	3.6.5
	FRMIDX	Frame index (1D), Array index (2D)	The address offset of frames within a block.	The address offset of arrays within a block.	3.6.5
20†	LINK	Link address	The PaRAM address containing the parameter set to be linked.		3.6.6
	ELERLD‡	Element count reload	The count value to be loaded at the end of each frame.‡	−−−	3.6.6

† Parameter set entries should always be accessed as 32-bit words using the STW or LDW instructions..
‡ This field is only valid for element-synchronized transfers.

3.3 Chaining EDMA Channels by an Event

See section 1.13 for an overview of chaining. Four of the user-specified 4-bit transfer complete codes (TCC values 8, 9, 10, and 11) can be used to trigger another EDMA channel transfer. These events trigger an EDMA transfer providing the ability to chain several EDMA transfer requests from one event that is driven by a peripheral or external device. By setting TCC to 8, 9, 10, or 11, any EDMA channel can synchronize any of these four channels.

To enable the EDMA controller to chain channels by way of a single event, you must set the TCINT bit to 1. Additionally, you should set the relevant bit in the channel chain enable register (CCER) to trigger off the next channel transfer specified by TCC. Since events 8−11 are the only EDMA channels that support chaining, only these bits are implemented in CCER. Reading unused bits returns the corresponding bits in EER and writing to the unused bits has no effect. Therefore, you can still specify a TCC value between 8−11, and not necessarily initiate the transfer on channels 8−11. However, ER bits 11−8 still captures the event, even if the corresponding bit in CCER is disabled. This allows selective enabling and disabling of these four specific events.

For example, if CCER[8] = 1 and TCC = 1000b are specified for EDMA channel 4, an external interrupt on EXT_INT4 initiates the EDMA transfer. Once the channel 4 transfer is complete (all of the parameters are exhausted), the EDMA controller initiates (TCINT = 1), the next transfer specified by EDMA channel 8. This is because TCC = 1000b is the sync event for EDMA channel 8. The corresponding CIPR bit 8 is set after channel 4 completes and generates an EDMA_INT (provided CIER[8] = 1) to the CPU. If the CPU interrupt is not desired, the corresponding interrupt enable bit (CIER[8]) must be cleared to 0. If the channel 8 transfer is not desired, CCER[8] must be cleared to 0.

The transfer complete code (TCC) is directly mapped to the CIPR bits as shown in Table 1−11.

3.4 EDMA Transfer Controller Priority

The EDMA channels have programmable priority. The PRI bits in the channel options parameter (OPT) specify the priority levels. The highest priority available is level 0 (urgent priority), which is not supported for EDMA channel controller transfer requests. Table 3−3 shows the available priority levels for the different requestors.

You should use care to not overburden the system by submitting all requests in high priority. Oversubscribing requests in one priority level can cause EDMA stalls and can be avoided by balanced bandwidth distribution in the different levels of priority. Refer to section 2.1.

Table 3−3. Programmable Priority Levels for Data Requests—C621x/C671x DSP

PRI Bits in OPT	Priority Level	Requestors
000	Level0; urgent priority	L2 controller
001	Level1; high priority	EDMA, QDMA and/or HPI
010	Level2; low priority	EDMA, QDMA
011−111	Reserved	Reserved

3.4.1 Transfer Controller Transfer Request Queue Length

The C621x/C671x EDMA transfer controller has three transfer request queues, Q0, Q1, and Q2, with the total number of transfer requests per queue as shown in Table 3−4. Transfer requests are sorted into Q0, Q1, and Q2, as based on the PRI field programmed into the requestor. The urgent priority queue, Q0, is reserved for L2 cache requests. The lower priority queues, Q1 and Q2, are used for EDMA channel controller, QDMA, and HPI transfers. Each queue can be shared by multiple requestors, but the number of entries allocated to each requestor is fixed, as shown in Table 3−4.

Table 3−4. Transfer Request Queues—C621x/C671x DSP

Queue	Priority Level	Total Queue Length (fixed)	Maximum Queue Length Available to Requester	
			Requestor	Queue Length
Q0	Level 0; urgent priority	6	L2 controller—cache requests	6
Q1	Level 1; high priority	13	EDMA	8
			L2 controller—QDMA requests	3
			HPI	2
Q2	Level 2; low priority	11	EDMA	8
			L2 controller—QDMA requests	3

3.4.1.1 Interaction Between Requestors

Each requestor individually tracks its queue allocation usage. Therefore, when a given requestor exceeds its allocation, only that requestor is impacted, as defined in the following sections. Note that L2 and QDMA are considered a single requestor.

3.4.1.2 EDMA Channel Controller Queue Allocation

EDMA channel requests should ideally be balanced between the high (Q1) and low (Q2) priority queues, such that the allocation on each queue is not exceeded. When an EDMA channel is processed but cannot be submitted due to the queue allocation for that queue being full, the EDMA controller stalls until a previously submitted EDMA channel controller transfer request exits that TC priority queue making room for the new request. While stalled, the EDMA controller cannot process any other events, including those events that could submit requests on a different priority queue. The EDMA event register (ER) still captures incoming events and the EDMA channel controller processes the pending events when the stall condition is released. Note that if the same event is asserted twice before being processed, the second event is dropped.

3.4.1.3 L2 Cache and QDMA Queue Allocation

Similar to the EDMA channel controller, if the L2 controller queue allocation is exceeded on a single priority level then other requests cannot be submitted until the staling request is able to proceed. This is not an issue for L2 controller cache requests, because it is assured that the L2 controller will never have more than six transfer requests outstanding at a given point in time and is the only possible requestor on Q0.

If back-to-back QDMAs are submitted such that the QDMAs allocation on a given priority level is exceeded, this will stall all other transfer requests (including QDMAs on different priority levels, and L2 cache requests). In addition, this has a potential to stall the entire CPU since the L1D and L2 controllers can be stalled waiting for the QDMA request to be submitted.

In order to avoid this situation, it is recommended that you limit the number of outstanding QDMA requests attempted on a given priority level to three. This can be done via inherent knowledge of the data flow of the system or can be manually tracked via software by using transfer completion code interrupts to track completion of QDMA requests.

3.4.1.4 HPI Queue Allocation

There is no issue with HPI queue allocation. It is assured that the HPI will never have more than two requests outstanding at a given point in time.

3.5 EDMA Control Registers

Each of the 16 channels in the EDMA has a specific synchronization event associated with it. These events trigger the data transfer associated with that channel. The list of control registers that perform various processing of events is shown in Table 3−5. These synchronization events are discussed in detail in section 1.5. See the device-specific data manual for the memory address of these registers and the event-to-channel mapping.

Table 3−5. EDMA Control Registers—C621x/C671x DSP

Acronym	Register Name	Section
ESEL†	EDMA event selector registers	3.5.1
PQSR	Priority queue status register	3.5.2
CIPR	EDMA channel interrupt pending register	3.5.3
CIER	EDMA channel interrupt enable register	3.5.4
CCER	EDMA channel chain enable register	3.5.5
ER	EDMA event register	3.5.6
EER	EDMA event enable register	3.5.7
ECR	EDMA event clear register	3.5.8
ESR	EDMA event set register	3.5.9

† See the device-specific data manual for availability.

3.5.1 EDMA Event Selector Registers (ESEL0, 1, 3)

The EDMA event selector registers (ESEL0, ESEL1, and ESEL3) control the mapping of the EDMA events to the EDMA channels on some C6000 devices (see your device-specific data manual for availability). The EDMA event selector registers are shown in Figure 3−2, Figure 3−3, and Figure 3−4 and described in Table 3−6, Table 3−7, and Table 3−8.

Each EDMA channel has a default EDMA event selector (EVTSEL) value, as listed in Table 3−9 (page 3-12). Each EDMA event selector (EVTSEL) value has an assigned EDMA event (see your device-specific data manual). Table 3−10 (page 3-12) shows the assigned EDMA event for the C6713 DSP. By loading each EVTSEL*n* field with an EDMA event selector value, you can map any desired EDMA event to any specified EDMA channel. For example on the C6713 DSP, if EVTSEL15 in ESEL3 is programmed to 00 0001b (EDMA selector value for TINT0), then EDMA channel 15 is triggered by Timer 0 TINT0 events.

Figure 3−2. EDMA Event Selector Register 0 (ESEL0)

31 30	29 24	23 22	21 16
Reserved	EVTSEL3	Reserved	EVTSEL2
R-0	R/W-03h	R-0	R/W-02h

15 14	13 8	7 6	5 0
Reserved	EVTSEL1	Reserved	EVTSEL0
R-0	R/W-01h	R-0	R/W-0

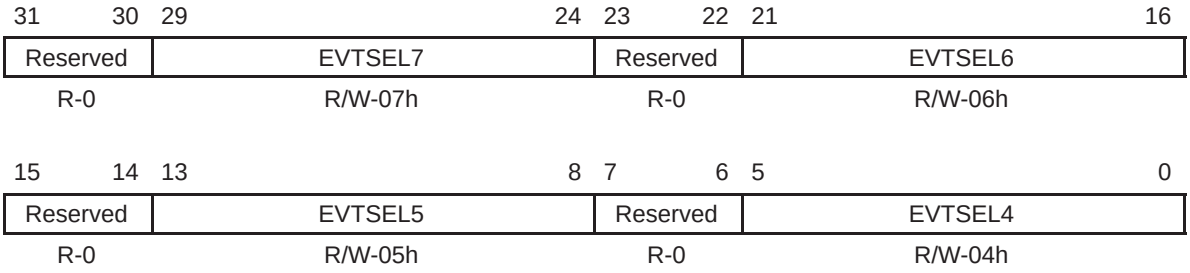
Legend: R = Read only, R/W = Read/Write; -n = value after reset

Table 3−6. EDMA Event Selector Register 0 (ESEL0) Field Descriptions

Bit	*field*†	*symval*†	Value	Description
31−30	Reserved	−	0	Reserved. You should always write 0 to this field.
29−24	EVTSEL3	OF(*value*)	0−3Fh	Event selector 3. This 6-bit value maps event 3 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	3h	EDMA channel 3 is triggered by SDINT event.
23−22	Reserved	−	0	Reserved. You should always write 0 to this field.
21−16	EVTSEL2	OF(*value*)	0−3Fh	Event selector 2. This 6-bit value maps event 2 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	2h	EDMA channel 2 is triggered by TINT1 event.
15−14	Reserved	−	0	Reserved. You should always write 0 to this field.
13−8	EVTSEL1	OF(*value*)	0−3Fh	Event selector 1. This 6-bit value maps event 1 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	1h	EDMA channel 1 is triggered by TINT0 event.
7−6	Reserved	−	0	Reserved. You should always write 0 to this field.
5−0	EVTSEL0	OF(*value*)	0−3Fh	Event selector 0. This 6-bit value maps event 0 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	0	EDMA channel 0 is triggered by DSPINT event.

† For CSL implementation, use the notation EDMA_ESEL0_*field_symval.*

Figure 3−3. EDMA Event Selector Register 1 (ESEL1)



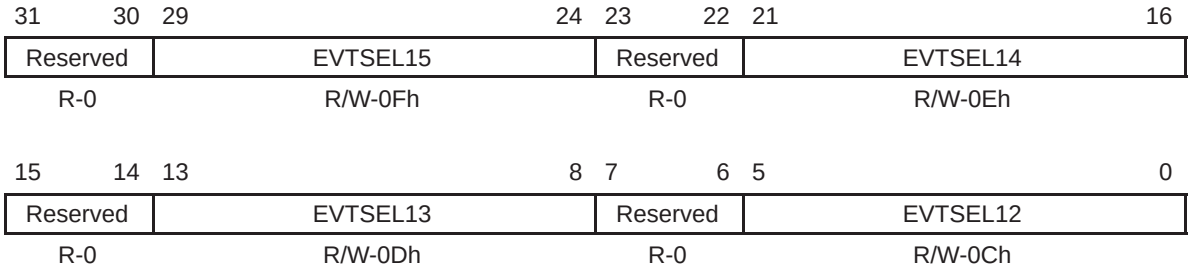
Legend: R = Read only, R/W = Read/Write; -n = value after reset

Table 3−7. EDMA Event Selector Register 0 (ESEL1) Field Descriptions

Bit	***field***†	***symval***†	Value	Description
31−30	Reserved	−	0	Reserved. You should always write 0 to this field.
29−24	EVTSEL7	OF(*value*)	0−3Fh	Event selector 7. This 6-bit value maps event 7 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	7h	EDMA channel 7 is triggered by EXTINT7 event.
23−22	Reserved	−	0	Reserved. You should always write 0 to this field.
21−16	EVTSEL6	OF(*value*)	0−3Fh	Event selector 6. This 6-bit value maps event 6 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	6h	EDMA channel 6 is triggered by EXTINT6 event.
15−14	Reserved	−	0	Reserved. You should always write 0 to this field.
13−8	EVTSEL5	OF(*value*)	0−3Fh	Event selector 5. This 6-bit value maps event 5 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	5h	EDMA channel 5 is triggered by EXTINT5 event.
7−6	Reserved	−	0	Reserved. You should always write 0 to this field.
5−0	EVTSEL4	OF(*value*)	0−3Fh	Event selector 4. This 6-bit value maps event 4 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	4h	EDMA channel 4 is triggered by EXTINT4 event.

† For CSL implementation, use the notation EDMA_ESEL1_*field_symval.*

Figure 3−4. EDMA Event Selector Register 3 (ESEL3)



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

Table 3−8. EDMA Event Selector Register 0 (ESEL3) Field Descriptions

Bit	***field***†	***symval***†	Value	Description
31−30	Reserved	−	0	Reserved. You should always write 0 to this field.
29−24	EVTSEL15	OF(*value*)	0−3Fh	Event selector 15. This 6-bit value maps event 15 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	Fh	EDMA channel 15 is triggered by REVT1 event.
23−22	Reserved	−	0	Reserved. You should always write 0 to this field.
21−16	EVTSEL14	OF(*value*)	0−3Fh	Event selector 14. This 6-bit value maps event 14 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	Eh	EDMA channel 14 is triggered by XEVT1 event.
15−14	Reserved	−	0	Reserved. You should always write 0 to this field.
13−8	EVTSEL13	OF(*value*)	0−3Fh	Event selector 13. This 6-bit value maps event 13 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	Dh	EDMA channel 13 is triggered by REVT0 event.
7−6	Reserved	−	0	Reserved. You should always write 0 to this field.
5−0	EVTSEL12	OF(*value*)	0−3Fh	Event selector 12. This 6-bit value maps event 12 to any EDMA channel. See Table 3−9 and Table 3−10.
		DEFAULT	Ch	EDMA channel 12 is triggered by XEVT0 event.

† For CSL implementation, use the notation EDMA_ESEL3_*field_symval.*

Table 3−9. Default EDMA Event for EDMA Channel—C6713 DSP

	EDMA Event Selector Register			
EDMA Channel	Bits	Field	Default EDMA Event Selector Value (Binary)	Default EDMA Event
0	ESEL0[5−0]	EVTSEL0	00 0000	DSPINT
1	ESEL0[13−8]	EVTSEL1	00 0001	TINT0
2	ESEL0[21−16]	EVTSEL2	00 0010	TINT1
3	ESEL0[29−24]	EVTSEL3	00 0011	SDINT
4	ESEL1[5−0]	EVTSEL4	00 0100	EXTINT4
5	ESEL1[13−8]	EVTSEL5	00 0101	EXTINT5
6	ESEL1[21−16]	EVTSEL6	00 0110	EXTINT6
7	ESEL1[29−24]	EVTSEL7	00 0111	EXTINT7
8	−	−	−	TCC8 (Chaining)
9	−	−	−	TCC9 (Chaining)
10	−	−	−	TCC10 (Chaining)
11	−	−	−	TCC11 (Chaining)
12	ESEL3[5−0]	EVTSEL12	00 1100	XEVT0
13	ESEL3[13−8]	EVTSEL13	00 1101	REVT0
14	ESEL3[21−16]	EVTSEL14	00 1110	XEVT1
15	ESEL3[29−24]	EVTSEL15	00 1111	REVT1

Table 3−10. EDMA Event for EDMA Event Selector Value—C6713 DSP

EDMA Event Selector Value (Binary)	EDMA Event	Module
00 0000	DSPINT	HPI
00 0001	TINT0	TIMER0
00 0010	TINT1	TIMER1
00 0011	SDINT	EMIF
00 0100	EXTINT4	GPIO
00 0101	EXTINT5	GPIO
00 0110	EXTINT6	GPIO
00 0111	EXTINT7	GPIO
00 1000	GPINT0	GPIO
00 1001	GPINT1	GPIO
00 1010	GPINT2	GPIO

Table 3−10. EDMA Event for EDMA Event Selector Value—C6713 DSP (Continued)

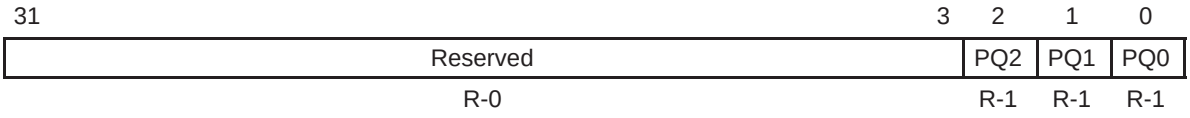
EDMA Event Selector Value (Binary)	EDMA Event	Module
00 1011	GPINT3	GPIO
00 1100	XEVT0	McBSP0
00 1101	REVT0	McBSP0
00 1110	XEVT1	McBSP1
00 1111	REVT1	McBSP1
01 0000−01 1111	Reserved	−
10 0000	AXEVTE0	McASP0
10 0001	AXEVTO0	McASP0
10 0010	AXEVT0	McASP0
10 0011	AREVTE0	McASP0
10 0100	AREVTO0	McASP0
10 0101	AREVT0	McASP0
10 0110	AXEVTE1	McASP1
10 0111	AXEVTO1	McASP1
10 1000	AXEVT1	McASP1
10 1001	AREVTE1	McASP1
10 1010	AREVTO1	McASP1
10 1011	AREVT1	McASP1
10 1100	I2CREVT0	I2C0
10 1101	I2CXEVT0	I2C0
10 1110	I2CREVT1	I2C1
10 1111	I2CXEVT1	I2C1
11 0000	GPINT8	GPIO
11 0001	GPINT9	GPIO
11 0010	GPINT10	GPIO
11 0011	GPINT11	GPIO
11 0100	GPINT12	GPIO
11 0101	GPINT13	GPIO
11 0110	GPINT14	GPIO
11 0111	GPINT15	GPIO
11 1000−11 1111	Reserved	−

3.5.2 Priority Queue Status Register (PQSR)

The priority queue status register (PQSR) indicates whether the transfer controller is empty on each priority level. The PQSR is shown in Figure 3−5 and described in Table 3−11. The priority queue status (PQ) bit provides the status of the queues as well as any active transfers. When the PQ bits are set to 111b, there are no requests pending in the respective priority level queues and no transfer is in progress. For example, if bit 0 (PQ0) is set to 1, all L2 requests for data movement have been completed and there are no requests pending in the priority level 0 queue.

The PQ bits are mainly used for emulation or debugging and typically should not be used by an application.

Figure 3−5. Priority Queue Status Register (PQSR)



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

Table 3−11. Priority Queue Status Register (PQSR) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−3	Reserved	−	0	Reserved. You should always write 0 to this field.
2−0	PQ	OF(*value*)	0−7h	Priority queue status. A 1 in the PQ bit indicates that there are no requests pending in the respective priority level queue.
		DEFAULT	7h	There are no requests pending in the priority level queues.

† For CSL implementation, use the notation EDMA_PQSR_PQ_*symval.*

3.5.3 EDMA Channel Interrupt Pending Register (CIPR)

The EDMA channel interrupt pending register (CIPR) is shown in Figure 3−6 and described in Table 3−12.

Figure 3−6. EDMA Channel Interrupt Pending Register (CIPR)

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

15	14	13	12	11	10	9	8
CIP15	CIP14	CIP13	CIP12	CIP11	CIP10	CIP9	CIP8
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

7	6	5	4	3	2	1	0
CIP7	CIP6	CIP5	CIP4	CIP3	CIP2	CIP1	CIP0
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

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

Table 3−12. EDMA Channel Interrupt Pending Register (CIPR) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−16	Reserved	−	0	Reserved. You should always write 0 to this field.
15−0	CIP	OF(*value*)	0−FFFFh	Channel interrupt pending. When the TCINT bit in the channel options parameter (OPT) is set to 1 for an EDMA channel and a specific transfer complete code (TCC) is provided by the EDMA transfer controller, the EDMA channel controller sets a bit in the CIP field.
		DEFAULT	0	EDMA channel interrupt is not pending.
		−	1	EDMA channel interrupt is pending.

† For CSL implementation, use the notation EDMA_CIPR_CIP_*symval.*

3.5.4 EDMA Channel Interrupt Enable Register (CIER)

The EDMA channel interrupt enable register (CIER) is shown in Figure 3−7 and described in Table 3−13.

Figure 3−7. EDMA Channel Interrupt Enable Register (CIER)

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

15	14	13	12	11	10	9	8
CIE15	CIE14	CIE13	CIE12	CIE11	CIE10	CIE9	CIE8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
CIE7	CIE6	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 3−13. EDMA Channel Interrupt Enable Register (CIER) Field Descriptions

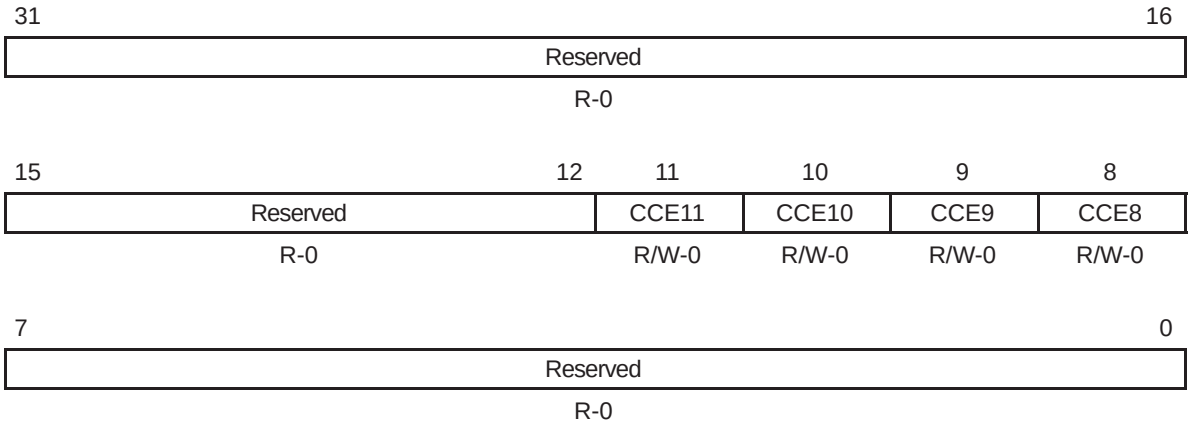
Bit	Field	*symval*†	Value	Description
31−16	Reserved	−	0	Reserved. You should always write 0 to this field.
15−0	CIE	OF(*value*)	0−FFFFh	Channel interrupt enable. A 16-bit unsigned value used to disable or enable an interrupt for an EDMA channel.
		DEFAULT	0	EDMA channel interrupt is not enabled.
		−	1	EDMA channel interrupt is enabled.

† For CSL implementation, use the notation EDMA_CIER_CIE_*symval*.

3.5.5 EDMA Channel Chain Enable Register (CCER)

The EDMA channel chain enable register (CCER) is shown in Figure 3−8 and described in Table 3−14.

Figure 3−8. EDMA Channel Chain Enable Register (CCER)



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

Table 3−14. EDMA Channel Chain Enable Register (CCER) Field Descriptions

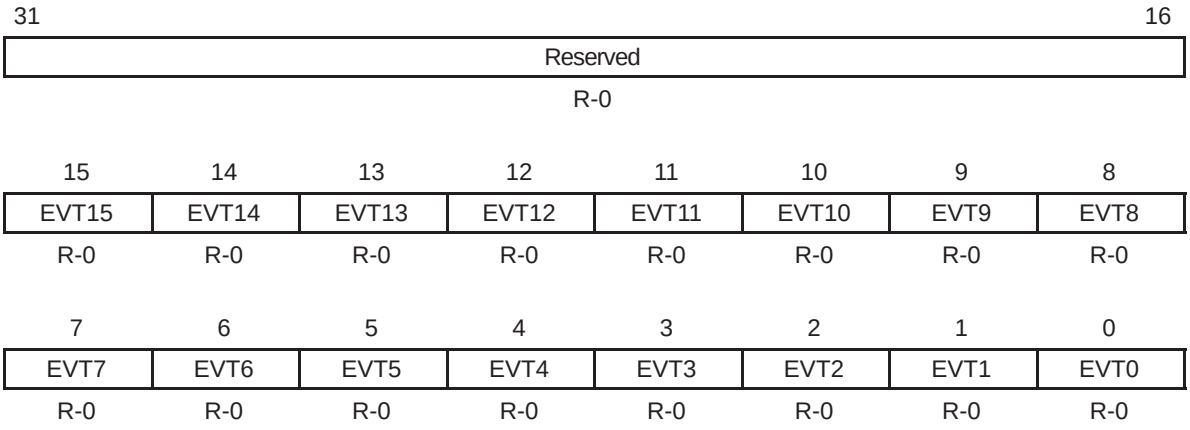
Bit	Field	*symval*†	Value	Description
31−12	Reserved	−	0	Reserved. You should always write 0 to this field.
11−8	CCE	OF(*value*)	0−Fh	Channel chain enable. To enable the EDMA controller to chain channels by way of a single event, set the TCINT bit in the channel options parameter (OPT) to 1. Additionally, set the relevant bit in the CCE field to trigger off the next channel transfer specified by TCC.
		DEFAULT	0	EDMA channel chain is not enabled.
		−	1	EDMA channel chain is enabled.
7−0	Reserved	−	0	Reserved. You should always write 0 to this field.

† For CSL implementation, use the notation EDMA_CCER_CCE_*symval.*

3.5.6 EDMA Event Register (ER)

The event register (ER) captures all events, even when the events are disabled. The ER is shown in Figure 3−9 and described in Table 3−15. Section 1.5 describes the type of synchronization events and the EDMA channels associated with each of them.

Figure 3−9. EDMA Event Register (ER)



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

Table 3−15. EDMA Event Register (ER) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−16	Reserved	−	0	Reserved. You should always write 0 to this field.
15−0	EVT	OF(*value*)	0−FFFFh	Event. All events that are captured by the EDMA are latched in ER, even if that event is disabled (EER = 0).
		DEFAULT	0	EDMA event is not asserted.
		−	1	EDMA event is asserted.

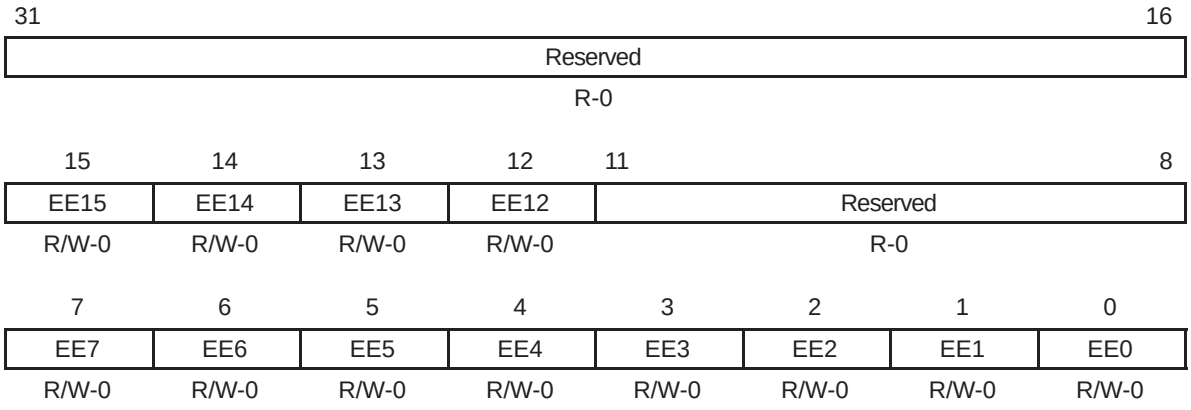
† For CSL implementation, use the notation EDMA_ER_EVT_*symval.*

3.5.7 EDMA Event Enable Register (EER)

Each event in the event register (ER) can be enabled or disabled using the event enable register (EER). Any of the event bits in EER can be set to 1 to enable that corresponding event or can be cleared to 0 to disable that corresponding event. The EER is shown in Figure 3−10 and described in Table 3−16.

Events 8−11 are only available for chaining of EDMA events; therefore, they are enabled in the channel chain enable register (CCER) and are not used in EER (these bits are reserved). The ER latches all events that are captured by the EDMA, even if that event is disabled. This is analogous to an interrupt enable and interrupt-pending register for interrupt processing, thus ensuring that the EDMA does not drop any events. Reenabling an event with a pending event signaled in ER forces the EDMA controller to process that event according to its priority.

Figure 3−10. EDMA Event Enable Register (EER)



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

Table 3−16. EDMA Event Enable Register (EER) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−16	Reserved	−	0	Reserved. You should always write 0 to this field.
15−0	EE	OF(*value*)	0−FFFFh	Event enable. Used to enable or disable an event. Note that bits 11−8 are only available for chaining of EDMA events; therefore, they are enabled in the channel chain enable register (CCER). Bits 11−8 are reserved and should only be written with 0.
		DEFAULT	0	EDMA event is not enabled.
		−	1	EDMA event is enabled.

† For CSL implementation, use the notation EDMA_EER_EE_*symval.*

3.5.8 Event Clear Register (ECR)

Once an event has been posted in the event register (ER), it can clear the event in two ways. If the event is enabled in the event enable register (EER) and the EDMA submits a transfer request for that event, it clears the corresponding event bit in ER. Alternatively, if the event is disabled in EER, the CPU can clear the event using the event clear register (ECR), shown in Figure 3−11 and described in Table 3−17.

Writing a 1 to any of the bits clears the corresponding event; writing a 0 has no effect. Once an event bit is set in ER, it remains set until the EDMA submits a transfer request for that event or the CPU clears the event by setting the corresponding bit in ECR.

Figure 3−11. EDMA Event Clear Register (ECR)

31							16
Reserved							
R-0							

15	14	13	12	11	10	9	8
EC15	EC14	EC13	EC12	EC11	EC10	EC9	EC8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 3−17. EDMA Event Clear Register (ECR) Field Descriptions

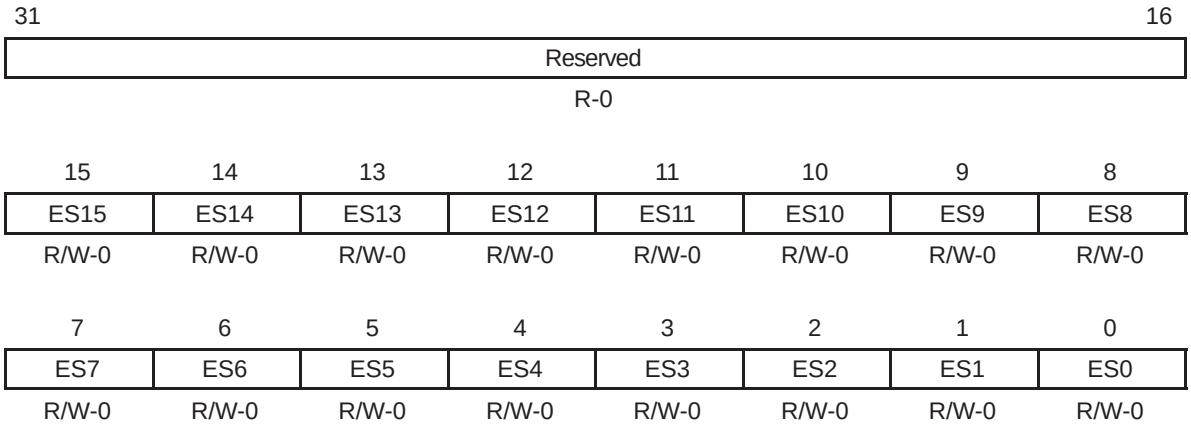
Bit	Field	*symval*†	Value	Description
31−16	Reserved	−	0	Reserved. You should always write 0 to this field.
15−0	EC	OF(*value*)	0−FFFFh	Event clear. Any of the event bits can be set to 1 to clear that event; a write of 0 has no effect.
		DEFAULT	0	No effect.
		−	1	EDMA event is cleared.

† For CSL implementation, use the notation EDMA_ECR_EC_*symval.*

3.5.9 EDMA Event Set Register (ESR)

The CPU can set events using the event set register (ESR), shown in Figure 3−12 and described in Table 3−18. Writing a 1 to one of the event bits causes a transfer request to be submitted. The event does not have to be enabled. This provides a debugging tool and also allows the CPU to submit EDMA requests in the system. Note that CPU-initiated EDMA transfers are considered CPU-synchronized transfers. In other words, an EDMA transfer occurs when the corresponding ESR bit is set and is not triggered by the associated event. See section 1.16 for a description of the quick DMA (QDMA), an alternative way to perform CPU-initiated EDMA transfers.

Figure 3−12. EDMA Event Set Register (ESR)



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

Table 3−18. EDMA Event Set Register (ESR) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−16	Reserved	−	0	Reserved. You should always write 0 to this field.
15−0	ES	OF(*value*)	0−FFFFh	Event set. Any of the event bits can be set to 1 to set the corresponding bit in the event register (ER); a write of 0 has no effect.
		DEFAULT	0	No effect.
		−	1	EDMA event is set.

† For CSL implementation, use the notation EDMA_ESR_ES_*symval.*

3.6 EDMA Channel Parameter Entries

See section 1.3 for an overview of the parameter RAM (PaRAM). Each parameter set of an EDMA channel is organized into six 32-bit words or 24 bytes as listed in Table 3−19. See Table 3−1 (page 3-3) for the memory address of these registers.

Table 3−19. EDMA Parameter Entries—C621x/C671x DSP

Acronym	Parameter Name	Section
OPT	EDMA channel options parameter	3.6.1
SRC	EDMA channel source address parameter	3.6.2
CNT	EDMA channel transfer count parameter	3.6.3
DST	EDMA channel destination address parameter	3.6.4
IDX	EDMA channel index parameter	3.6.5
RLD	EDMA channel count reload/link address parameter	3.6.6

3.6.1 EDMA Channel Options Parameter (OPT)

The EDMA channel options parameter (OPT) in the EDMA parameter entries is shown in Figure 3−13 and described in Table 3−20.

Figure 3−13. EDMA Channel Options Parameter (OPT)

31 29	28 27	26	25 24	23	22 21	20	19 16
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x

15 2	1	0
Reserved†	LINK	FS
R/W-0	R/W-x	R/W-x

† Always write 0 to the reserved bits.
Legend: R/W = Read/Write; -x = value is indeterminate after reset

Table 3−20. EDMA Channel Options Parameter (OPT) Field Descriptions

Bit	*field*†	*symval*†	Value	Description
31−29	PRI	OF(*value*)	0−7h	Priority levels for EDMA events.
		DEFAULT	0	Reserved. This level is reserved only for L2 requests and not valid for EDMA channel or QDMA transfer requests.
		HIGH	1h	High priority EDMA transfer.
		LOW	2h	Low priority EDMA transfer.
		−	3h−7h	Reserved.
28−27	ESIZE	OF(*value*)	0−3h	Element size.
		DEFAULT 32BIT	0	32-bit word.
		16BIT	1h	16-bit halfword.
		8BIT	2h	8-bit byte.
		−	3h	Reserved.
26	2DS	OF(*value*)		Source dimension.
		DEFAULT NO	0	1-dimensional source.
		YES	1	2-dimensional source.
25−24	SUM	OF(*value*)	0−3h	Source address update mode.
		DEFAULT NONE	0	Fixed address mode. No source address modification.
		INC	1h	Source address increment depends on the 2DS and FS bits.
		DEC	2h	Source address decrement depends on the 2DS and FS bits.
		IDX	3h	Source address modified by the element index/frame index depending on the 2DS and FS bits.
23	2DD	OF(*value*)		Destination dimension.
		DEFAULT NO	0	1-dimensional destination.
		YES	1	2-dimensional destination.

† For CSL implementation, use the notation EDMA_OPT_*field_symval.*

Table 3−20. EDMA Channel Options Parameter (OPT) Field Descriptions (Continued)

Bit	*field*†	*symval*†	Value	Description
22−21	DUM	OF(*value*)	0−3h	Destination address update mode.
		DEFAULT NONE	0	Fixed address mode. No destination address modification.
		INC	1h	Destination address increment depends on the 2DD and FS bits.
		DEC	2h	Destination address decrement depends on the 2DD and FS bits.
		IDX	3h	Destination address modified by the element index/frame index depending on the 2DD and FS bits.
20	TCINT	OF(*value*)		Transfer complete interrupt.
		DEFAULT NO	0	Transfer complete indication is disabled. The EDMA channel interrupt pending register (CIPR) bits are not set upon completion of a transfer.
		YES	1	Transfer complete indication is enabled. The EDMA channel interrupt pending register (CIPR) bit is set on a channel transfer completion. The bit (position) set in CIPR is the TCC value specified. This bit can be used for chaining and interrupt generation.
19−16	TCC	OF(*value*)	0−Fh	Transfer complete code. This 4-bit value is used to set the bit in the EDMA channel interrupt pending register (CIPR[TCC] bit) provided. This bit can be used for chaining and interrupt generation.
		DEFAULT	0	
15−2	Reserved	−	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. If writing to this field, always write a 0.
1	LINK	OF(*value*)		Linking of event parameters enable.
		DEFAULT NO	0	Linking of event parameters is disabled. Entry is not reloaded.
		YES	1	Linking of event parameters is enabled. After the current set is exhausted, the channel entry is reloaded with the parameter set specified by the link address.

† For CSL implementation, use the notation EDMA_OPT_*field_symval.*

Table 3−20. EDMA Channel Options Parameter (OPT) Field Descriptions (Continued)

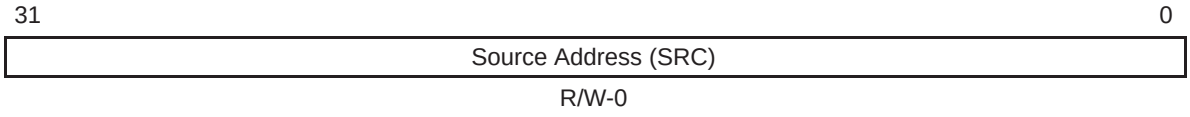
Bit	*field*†	*symval*†	Value	Description
0	FS	OF(*value*)		Frame synchronization.
		DEFAULT	0	Channel is element/array synchronized.
		NO		
		YES	1	Channel is frame synchronized. The relevant event for a given EDMA channel is used to synchronize a frame.

† For CSL implementation, use the notation EDMA_OPT_*field_symval.*

3.6.2 EDMA Channel Source Address Parameter (SRC)

The EDMA channel source address parameter (SRC) in the EDMA parameter entries specifies the starting byte address of the source. The SRC is shown in Figure 3−14 and described in Table 3−21. Use the SUM bits in the EDMA channel options parameter (OPT) to modify the source address. See section 1.9.2 for details. The source address must be aligned to the value specified by ESIZE, refer to section 1.7.

Figure 3−14. EDMA Channel Source Address Parameter (SRC)



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

Table 3−21. EDMA Channel Source Address Parameter (SRC) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	SRC	OF(*value*)	0−FFFF FFFFh	This 32-bit source address specifies the starting byte address of the source. The address is modified using the SUM bits in the EDMA channel options parameter (OPT).
		DEFAULT	0	

† For CSL implementation, use the notation EDMA_SRC_SRC_*symval.*

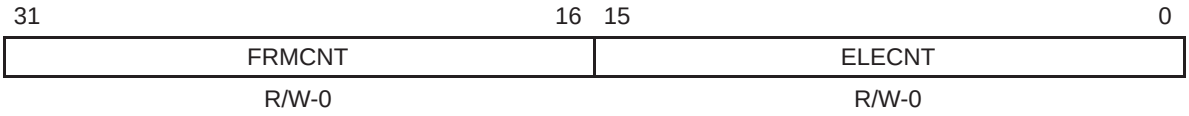
3.6.3 EDMA Channel Transfer Count Parameter (CNT)

The EDMA channel transfer count parameter (CNT) in the EDMA parameter entries specifies the frame/array count and element count. The CNT is shown in Figure 3−15 and described in Table 3−22.

The frame/array count (FRMCNT) is a 16-bit unsigned value plus 1 that specifies the number of frames in a 1D block or the number of arrays in a 2D block. Frame count applies to 1D transfers and array count applies to 2D transfers. Valid values for the frame/array count range between 0 to 65535; therefore, the maximum number of frames/arrays in a block is 65536. A frame/array count of 0 is actually one frame/array, and a frame/array count of 1 is two frames/arrays. See section 1.9.1 for details.

The element count (ELECNT) is a 16-bit unsigned value that specifies the number of elements in a frame (for 1D transfers) or in an array (for 2D transfers). Valid values for the element count range between 1 to 65535; therefore, the maximum number of elements in a frame is 65535. The EDMA performs no transfers if the element count is 0. See section 1.9.1 for details.

Figure 3−15. EDMA Channel Transfer Count Parameter (CNT)



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

Table 3−22. EDMA Channel Transfer Count Parameter (CNT) Field Descriptions

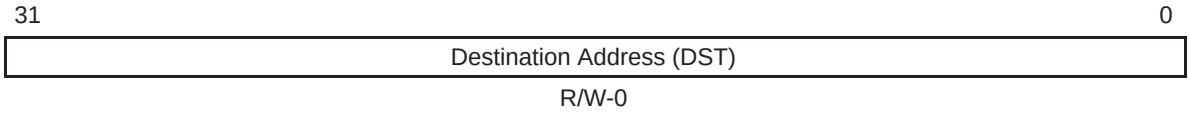
Bit	*field*†	*symval*†	Value	Description
31−16	FRMCNT	OF(*value*)	0−FFFFh	Frame/array count. A 16-bit unsigned value plus 1 that specifies the number of frames in a 1D block or number of arrays in a 2D block. Valid values for the frame/array count: 0−65535.
		DEFAULT	0	1 frame in a 1D block or 1 array in a 2D block.
15−0	ELECNT	OF(*value*)	0−FFFFh	Element count. A 16-bit unsigned value that specifies the number of elements in a frame (for 1D transfers) or an array (for 2D transfers). Valid values for the element count: 1−65535.
		DEFAULT	0	No transfer.

† For CSL implementation, use the notation EDMA_CNT_*field_symval.*

3.6.4 EDMA Channel Destination Address Parameter (DST)

The EDMA channel destination address parameter (DST) in the EDMA parameter entries specifies the starting byte address of the destination. The DST is shown in Figure 3−16 and described in Table 3−23. Use the DUM bits in the EDMA channel options parameter (OPT) to modify the destination address. See section 1.9.2 for details. The destination address must be aligned to the value specified by ESIZE, refer to section 1.7.

Figure 3−16. EDMA Channel Destination Address Parameter (DST)



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

Table 3−23. EDMA Channel Destination Address Parameter (DST) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	DST	OF(*value*)	0−FFFF FFFFh	This 32-bit destination address specifies the starting byte address of the destination. The address is modified using the DUM bits in the EDMA channel options parameter (OPT).
		DEFAULT	0	

† For CSL implementation, use the notation EDMA_DST_DST_*symval.*

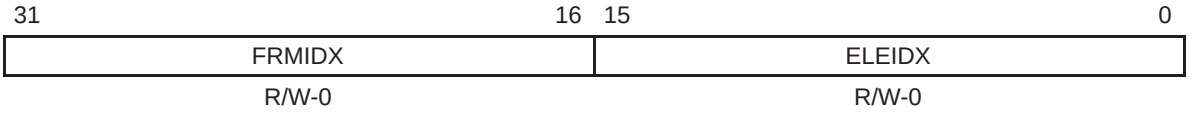
3.6.5 EDMA Channel Index Parameter (IDX)

The EDMA channel index parameter (IDX) in the EDMA parameter entries specifies the frame/array index and element index used for address modification. The EDMA is shown in Figure 3−17 and described in Table 3−24. The EDMA uses the indexes for address updates, depending on the type of transfer (1D or 2D) selected, and the FS, SUM, and DUM bits in the EDMA channel options parameter (OPT).

The frame/array index (FRMIDX) is a 16-bit signed value that specifies the address offset (in bytes) to the next frame/array in a 1D transfer or 2D transfer. Frame index applies to 1D transfers and array index applies to 2D transfers. Valid values for the frame/array index range between –32768 to 32767.

The element index (ELEIDX) is a 16-bit signed value that specifies the address offset (in bytes) to the next element in a frame. The element index is used only for 1D transfers, because 2D transfers do not allow spacing between elements. Valid values for the element index range between –32768 to 32767.

Figure 3−17. EDMA Channel Index Parameter (IDX)



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

Table 3−24. EDMA Channel Index Parameter (IDX) Field Descriptions

Bit	***field***†	***symval***†	Value	Description
31−16	FRMIDX	OF(*value*)	0−FFFFh	Frame/array index. A 16-bit signed value that specifies the frame/array index used for an address offset to the next frame/array. Valid values for the frame/array index: –32768 to 32767.
		DEFAULT	0	No offset is used as the frame/array index for an address offset to the next frame/array.
15−0	ELEIDX	OF(*value*)	0−FFFFh	Element index. A 16-bit signed value that specifies the element index used for an address offset to the next element in a frame. Element index is used *only* for 1D transfers. Valid values for the element index: –32768 to 32767.
		DEFAULT	0	No offset is used as the element index for an address offset to the next element in a frame.

† For CSL implementation, use the notation EDMA_IDX_*field_symval.*

3.6.6 EDMA Channel Count Reload/Link Address Parameter (RLD)

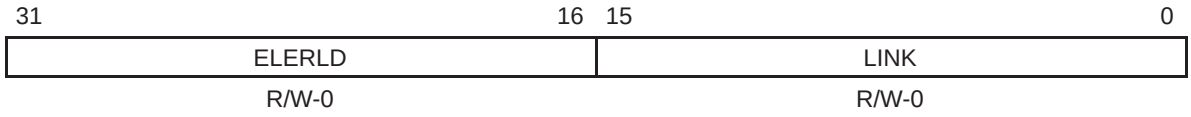
The EDMA channel count reload/link address parameter (RLD) in the EDMA parameter entries specifies the value used to reload the element count field and the link address. The RLD is shown in Figure 3−18 and described in Table 3−25.

The 16-bit unsigned element count reload (ELERLD) value reloads the element count (ELECNT) field in the EDMA channel transfer count parameter (CNT), once the last element in a frame is transferred. ELERLD is used only for a 1D element sync (FS = 0) transfer, since the EDMA has to keep track of the next element address using the element count. This is necessary for multi-frame EDMA transfers, where frame count value is greater than 0. See section 1.9.1 for more details.

The EDMA controller provides a mechanism to link EDMA transfers for autoinitialization. When LINK = 1 in the EDMA channel options parameter (OPT), the 16-bit link address (LINK) specifies the lower 16-bit address in the parameter RAM where the EDMA loads/reloads the parameters in preparation for the next event. Since the entire EDMA parameter RAM is located in the 01A0 xxxxh area, only the lower 16-bit address is required.

The reload parameters are specified in the address range starting at 01A0 0180h. You must ensure that the link address is on a 24-byte boundary, and that the operation is undefined if this rule is violated (see section 1.11). In addition to the reload parameter space, the entry of any unused EDMA channel can also be used for linking. The EDMA can always have up to 85 programmed entries, regardless of the number of channels actually used.

Figure 3−18. EDMA Channel Count Reload/Link Address Parameter (RLD)



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

Table 3−25. EDMA Channel Count Reload/Link Address Parameter (RLD) Field Descriptions

Bit	***field***†	***symval***†	Value	Description
31−16	ELERLD	OF(*value*)	0−FFFFh	Element count reload. A 16-bit unsigned value used to reload the element count field in the EDMA channel transfer count parameter (CNT) once the last element in a frame is transferred. This field is used only for a 1D element sync (FS = 0) transfer, since the EDMA has to keep track of the next element address using the element count. This is necessary for multi-frame EDMA transfers where frame count value is greater than 0.
		DEFAULT	0	0 is used to reload the element count field in the EDMA channel transfer count parameter (CNT) once the last element in a frame is transferred.
15−0	LINK	OF(*value*)	0−FFFFh	This 16-bit link address specifies the lower 16-bit address in the parameter RAM from which the EDMA loads/reloads the parameters of the next event in the chain.
		DEFAULT	0	The address 01A0 0000h in the parameter RAM is used to load/reload the parameters of the next event in the chain.

† For CSL implementation, use the notation EDMA_RLD_*field_symval.*

3.7 QDMA Registers

Since the QDMA is used for quick, one-time transfers it does not have the capability to reload a count or link. The count reload/link address register is therefore not available to the QDMA. The QDMA registers are not updated during or after a transfer by the hardware, they retain the submitted values. All QDMA transfers are submitted using frame synchronization (1D) or block synchronization (2D). See section 1.16 for more details.

The QDMA consists of two sets of memory-mapped, write-only registers (Figure 3−19), similar to an EDMA parameter entry. Reads of the QDMA registers return invalid data. The first set shown in Figure 3−19(a) is a direct mapping of the five QDMA registers required to configure a transfer. There is no count reload, no link address, and the LINK field of the QDMA channel options register (QOPT) is reserved. Writing to the QDMA registers configures, but does not submit, a QDMA transfer request. Figure 3−19(b) shows the pseudo-registers for this set. Writing to any of the pseudo-registers submits a transfer request.

Although the QDMA mechanism does not support event linking, it supports completion interrupts, as well as QDMA transfer-complete chaining to EDMA channels. QDMA completion interrupts are enabled and set in the same way as EDMA completion interrupts; through the use of the TCINT and TCC bits in the QDMA channel options register (QOPT), and CIPR and CIER of the EDMA. QDMA transfer-complete chaining with EDMA events are enabled through setting the appropriate bits in QOPT and CCER of the EDMA channel controller. QDMA transfer requests have the same priority restrictions as the EDMA. See section 3.4 for details.

Access to each register is limited to 32-bits only. Halfword and byte writes result in undefined behavior.

Figure 3−19. QDMA Registers

(a) QDMA registers

Address			QDMA register
0200 0000h	QDMA Channel Options		QOPT
0200 0004h	QDMA Channel Source Address (SRC)		QSRC
0200 0008h	Array/frame count (FRMCNT)	Element count (ELECNT)	QCNT
0200 000Ch	QDMA Channel Destination Address (DST)		QDST
0200 0010h	Array/frame index (FRMIDX)	Element index (ELEIDX)	QIDX

(b) QDMA pseudo-registers

Address			QDMA pseudo-register
0200 0020h	QDMA Channel Options		QSOPT
0200 0024h	QDMA Channel Source Address (SRC)		QSSRC
0200 0028h	Array/frame count (FRMCNT)	Element count (ELECNT)	QSCNT
0200 002Ch	QDMA Channel Destination Address (DST)		QSDST
0200 0030h	Array/frame index (FRMIDX)	Element index (ELEIDX)	QSIDX

3.7.1 QDMA Channel Options Register (QOPT, QSOPT)

Figure 3−20. QDMA Channel Options Register (QOPT)

31–29	28–27	26	25–24	23	22–21	20	19–16
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

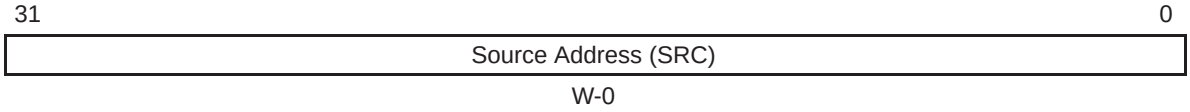
15–1	0
Reserved	FS
R/W-0	R/W-0

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

Note: QOPT is read/writable; QSOPT is write-only.

3.7.2 QDMA Channel Source Address Register (QSRC, QSSRC)

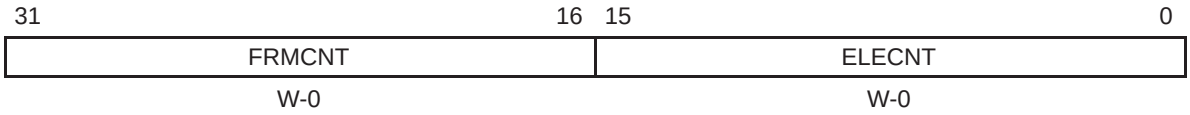
Figure 3−21. QDMA Channel Source Address Register (QSRC)



Legend: W= Write only; -*n* = value after reset

3.7.3 QDMA Channel Transfer Count Register (QCNT, QSCNT)

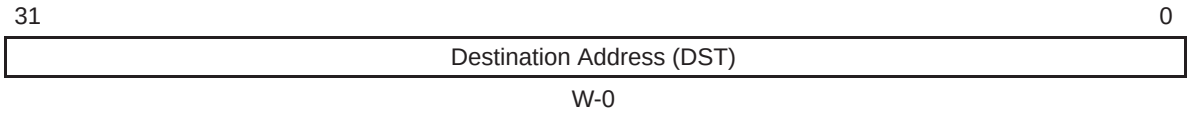
Figure 3−22. QDMA Channel Transfer Count Register (QCNT)



Legend: W= Write only; -*n* = value after reset

3.7.4 QDMA Channel Destination Address Register (QDST, QSDST)

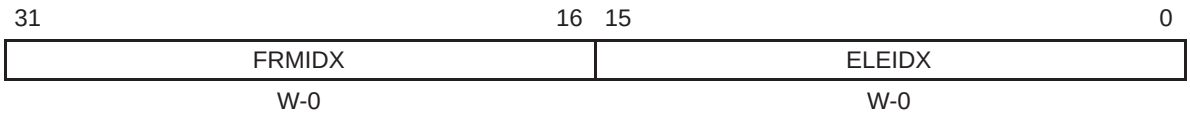
Figure 3−23. QDMA Channel Destination Address Register (QDST)



Legend: W= Write only; -*n* = value after reset

3.7.5 QDMA Channel Index Register (QIDX, QSIDX)

Figure 3−24. QDMA Channel Index Register (QIDX)



Legend: W= Write only; -*n* = value after reset

Chapter 4

TMS320C64x EDMA

This chapter describes the operation and registers of the EDMA controller in the TMS320C64x™ DSP. This chapter also describes the quick DMA (QDMA) registers that the CPU uses for fast data requests. For operation and registers unique to the TMS320C621x/C671x EDMA, see Chapter 3.

Topic **Page**

4.1 Event Service Priority

The EDMA event registers (ERL and ERH) capture up to 64 events; therefore, it is possible for events to occur simultaneously on the EDMA event inputs. For events arriving simultaneously, the channel with the highest event number submits its transfer request first. This mechanism only sorts simultaneous events and does nothing to set the priority of the events. The priority of the transfer is determined by its EDMA parameters stored in the parameter RAM (PaRAM) of the EDMA.

4.2 Parameter RAM (PaRAM)

Unlike the C620x/C670x DMA controller, which is a register-based architecture, the EDMA controller is a RAM-based architecture. EDMA channels are configured in a parameter table. The table is a block of parameter RAM (PaRAM) located within the EDMA channel controller. The total PaRAM size is device specific, ranging from 2-KByte to 5-KByte. Refer to the device-specific data manual for details. The PaRAM table (Table 4−1) consists of six-word parameter entries of 24 bytes each. The contents of the PaRAM include:

- 64 transfer parameter set entries for the 64 EDMA events. Each parameter set is six words or 24 bytes. These areas can also serve as reload/link parameters.
- Remaining parameter sets serve as additional parameters used for reloading or linking transfers. Each reload/link parameter set is 24 bytes. The total reload/link parameter sets is device specific.

Each parameter entry of an EDMA event is organized into six 32-bit words or 24 bytes, as shown in Figure 4−1 and described in Table 4−2.

Table 4−1. EDMA Parameter RAM Contents—C64x DSP

Address	Parameters
01A0 0000h to 01A0 0017h	Parameters for event 0 (6 words)
01A0 0018h to 01A0 002Fh	Parameters for event 1 (6 words)
01A0 0030h to 01A0 0047h	Parameters for event 2 (6 words)
01A0 0048h to 01A0 005Fh	Parameters for event 3 (6 words)
01A0 0060h to 01A0 0077h	Parameters for event 4 (6 words)
01A0 0078h to 01A0 008Fh	Parameters for event 5 (6 words)
01A0 0090h to 01A0 00A7h	Parameters for event 6 (6 words)
01A0 00A8h to 01A0 00BFh	Parameters for event 7 (6 words)
01A0 00C0h to 01A0 00D7h	Parameters for event 8 (6 words)
01A0 00D8h to 01A0 00EFh	Parameters for event 9 (6 words)
01A0 00F0h to 01A0 0107h	Parameters for event 10 (6 words)
01A0 0108h to 01A0 011Fh	Parameters for event 11 (6 words)
01A0 0120h to 01A0 0137h	Parameters for event 12 (6 words)
01A0 0138h to 01A0 014Fh	Parameters for event 13 (6 words)
01A0 0150h to 01A0 0167h	Parameters for event 14 (6 words)
01A0 0168h to 01A0 017Fh	Parameters for event 15 (6 words)
01A0 0180h to 01A0 0197h	Parameters for event 16 (6 words)
01A0 0198h to 01A0 01AFh	Parameters for event 17 (6 words)
...	...
01A0 05D0h to 01A0 05E7h	Parameters for event 62 (6 words)
01A0 05E8h to 01A0 05FFh	Parameters for event 63 (6 words)
01A0 0600h to 01A0 0617h	Reload/link parameter for Event 0 (6 words)
01A0 0618h to 01A0 062Fh	Reload/link parameter for Event 1 (6 words)
...	...
01A0 07E0h to 01A0 07F7h	Reload/link parameter for Event 20 (6 words)†

† The C6411/C6414/C6415/C6416 DSP has 21 parameter sets [6-words each] that can be used to reload/link EDMA transfers.
‡ All other C64x and DM64x DSPs have 149 parameter sets [6-words each] that can be used to reload/link EDMA transfers.

Table 4−1. EDMA Parameter RAM Contents—C64x DSP (Continued)

Address	Parameters
01A0 07F8h to 01A0 07FFh	Scratch pad area (2 words)†‡
01A0 07F8h to 01A0 080Fh	Reload/link parameter for Event 21 (6 words)‡
01A0 0810h to 01A0 0827h	Reload/link parameter for Event 22 (6 words)‡
...	...
01A0 13C8h to 01A0 13DFh	Reload/link parameter for Event 147 (6 words)‡
01A0 13E0h to 01A0 13F7h	Reload/link parameter for Event 148 (6 words)‡
01A0 13F8h to 01A0 13FFh	Scratch pad area (2 words)‡

† The C6411/C6414/C6415/C6416 DSP has 21 parameter sets [6-words each] that can be used to reload/link EDMA transfers.
‡ All other C64x and DM64x DSPs have 149 parameter sets [6-words each] that can be used to reload/link EDMA transfers.

Figure 4−1. EDMA Channel Parameter Entries for Each EDMA Event—C64x DSP

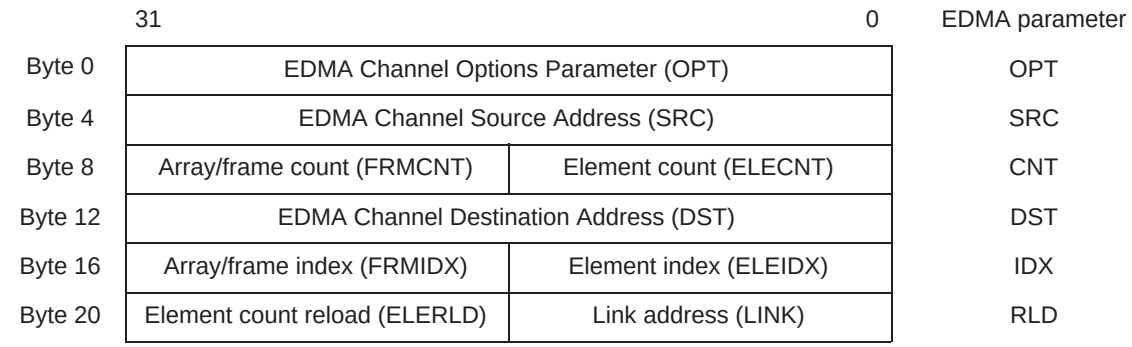


Table 4−2. EDMA Channel Parameter Descriptions—C64x DSP

Offset address (bytes)	Acronym	Parameter	As defined for…		Section
			1D transfer	2D transfer	
0	OPT	Channel options	Transfer configuration options.		4.8.1
4	SRC	Channel source address	The address from which data is transferred.		4.8.2
8†	ELECNT	Element count	The number of elements per frame.	The number of elements per array.	4.8.3
	FRMCNT	Frame count (1D), Array count (2D)	The number of frames per block minus 1.	The number of arrays per block minus 1.	4.8.3
12	DST	Channel destination address	The address to which data is transferred.		4.8.4
16†	ELEIDX	Element index	The address offset of elements within a frame.	−−−	4.8.5
	FRMIDX	Frame index (1D), Array index (2D)	The address offset of frames within a block.	The address offset of arrays within a block.	4.8.5
20†	LINK	Link address	The PaRAM address containing the parameter set to be linked.		4.8.6
	ELERLD	Element count reload	The count value to be loaded at the end of each frame.‡	−−−	4.8.6

† Parameter set entries should always be accessed as 32-bit words using the STW or LDW instructions..
‡ This field is only valid for element-synchronized transfers.

4.3 Chaining EDMA Channels With a Single Event

See section 1.13 for an overview of chaining. Any of the 64 transfer completion codes of the EDMA can trigger another channel transfer. To enable the EDMA controller to chain channels by way of a single event, you must set the TCINT bit to 1. Additionally, you should set the relevant bit in the channel chain enable register (CCERL or CCERH) to trigger off the next channel transfer specified by the transfer complete code.

The user-specified transfer complete code is expanded to a 6-bit value, TCCM:TCC. The 4-bit TCC field in the channel options parameter (OPT) is the least-significant bits of the transfer complete code and the 2-bit TCCM field in OPT is the most-significant bits of the transfer complete code. For example, if the transfer complete code (TCCM:TCC) is 01 0001b (TCCM = 01 and TCC = 0001) and CCERL[17] = 1 is specified for EDMA channel 4, the completion of the channel 4 transfer initiates the next transfer specified by EDMA channel 17, provided that the channel 4 TCINT = 1.

The transfer complete code (TCC) is directly mapped to the CIPR bits as shown in Table 1−12.

4.3.1 Alternate Transfer Chaining

The alternate transfer complete interrupt (ATCINT) bit and alternate transfer complete code (ATCC) bits in the channel options parameter (OPT) allow the C64x EDMA to perform channel chaining upon completion of intermediate transfers in a block. The function of the alternate transfer chaining is similar to the function of the transfer complete chaining. Alternate transfer complete code chaining does not affect linking operations described in section 1.11.

When alternate transfer complete chaining is enabled, the next EDMA channel (specified by the ATCC value of the current channel) is synchronized upon completion of each intermediate transfer of the current channel. Upon completion of the entire channel transfer, transfer complete chaining applies instead, provided transfer complete chaining is enabled. Alternate transfer complete chaining does not apply to 2D block-synchronized transfers, since this mode does not have intermediate transfers. Alternate transfer chaining allows one channel to trigger another channel once for each transfer request it makes (once per sync event received), rather than only once per block.

To enable alternate transfer complete chaining, configure the EDMA channel parameter as follows:

- ❑ Set ATCINT = 1, in OPT
- ❑ Set ATCC value to the next EDMA channel in the chain
- ❑ Set the relevant bit in the channel chain enable register (CCER[ATCC])

4.4 Peripheral Device Transfers

The C64x EDMA supports the peripheral device transfer mode (PDT), which provides an efficient way to transfer large amounts of data between an external peripheral device and an external memory device that share the same data pins. In normal operation, this type of transfer requires an EMIF read of the external source followed by an EMIF write to the external destination requiring 2 EMIF bus cycles. When PDT is enabled, data is driven by the external source directly, and written to the external destination in the same data bus transaction. See *TMS320C6000 DSP External Memory Interface (EMIF) Reference Guide* (SPRU266) for a detailed description of PDT.

PDT transfers are classified in terms of the memory on the EMIF. A PDT write is a transfer from a peripheral to memory (memory is physically written). To enable a PDT write from an external peripheral source to an external memory destination, set the PDTD bit in the channel options parameter (OPT) to 1.

A PDT read is a transfer from memory to a peripheral (memory is physically read). To enable a PDT read from an external memory source to an external peripheral destination, set the PDTS bit in OPT to 1.

PDT writes and PDT reads are mutually exclusive. In other words, PDTS and PDTD cannot both be set to 1.

When performing a PDT transfer, the source address and destination address must be set to the same value.

4.5 EDMA Transfer Controller Priority

The EDMA channels have programmable priority. The PRI bits in the channel options parameter (OPT) specify the priority levels. The highest priority available is level 0 (urgent priority). Table 4−3 shows the available priority levels for the different requestors.

You should use care to not overburden the system by submitting all requests in high priority. Since requests on a single priority level are serviced serially and in order; whereas, requests on different priority levels can be serviced concurrently. Oversubscribing requests in one priority level can cause EDMA stalls and can be avoided by balanced bandwidth distribution in the different levels of priority. Refer to section 2.1.

Table 4−3. Programmable Priority Levels for Data Requests—C64x DSP

PRI Bits in OPT	Priority Level	Requestors
000	Level 0; urgent priority	Any
001	Level 1; high priority	Any
010	Level 2; medium priority	Any
011	Level 3, low priority	Any
100−111	Reserved	Reserved

4.5.1 Transfer Controller Transfer Request Queue Length

The C64x EDMA transfer controller has four transfer request queues, Q0, Q1, Q2, and Q3, with a fixed length of 16 transfer requests per queue. Table 4−4 shows the transfer request queues for the C6416 device. Refer to the device-specific data manual for the transfer request queues of other devices. Transfer requests are sorted into Q0, Q1, Q2, and Q3, as based on the PRI field programmed into the requestor. There are no hard-coded restrictions on which queues can be associated with which requestors. Each queue can be shared by multiple requestors and the number of entries allocated to each requestor, as well as the queue usage for each requestor is programmable using the register listed in Table 4−4.

Table 4−4. Transfer Request Queues—C6416 DSP

Queue	Priority Level (PRI)	Total Queue Length (fixed)	Requestor†	Default Queue Length	Register to Program Queue Length
Q0	0; urgent priority	16	L2 controller and QDMA	6	L2ALLOC0
			EDMA	2	PQAR0
			HPI/PCI	0	TRCTL
Q1	1; high priority	16	L2 controller and QDMA	2	L2ALLOC1
			EDMA	6	PQAR1
			HPI/PCI	0	TRCTL
Q2	2; medium priority	16	L2 controller and QDMA	2	L2ALLOC2
			EDMA	2	PQAR2
			HPI/PCI	4	TRCTL
Q3	3; low priority	16	L2 controller and QDMA	2	L2ALLOC3
			EDMA	6	PQAR3
			HPI/PCI	0	TRCTL

† L2 controller and QDMA share one queue allocation. L2ALLOC*n* register controls this queue allocation length.
HPI and PCI share one queue allocation.
TRCTL register in the HPI module controls the queue allocation length of HPI requests.
TRCTL register in the PCI module controls the queue allocation length of PCI requests.
Refer to the data manual for specific queue allocation.

4.5.1.1 Interaction Between Requestors

Each requestor individually tracks its queue allocation usage, which can be user programmed. Therefore, when a given requestor exceeds its allocation, only that requestor is impacted, as defined in the following sections. Note that L2 and QDMA are considered a single requestor.

4.5.1.2 EDMA Channel Controller Queue Allocation

EDMA channel requests should ideally be balanced between each of the four priority queues such that the allocation on each queue is not exceeded. When an EDMA channel is processed but cannot be submitted due to the queue allocation for that queue being full, the EDMA channel controller stalls until a previously submitted EDMA channel controller transfer request exits that TC priority queue making room for the new request. While stalled, the EDMA channel controller cannot process any other events, including those events that could submit requests on a different priority queue. The EDMA event register (ER) still captures incoming events and the EDMA channel controller processes the pending events when the stall condition is released. Note that if the same event is asserted twice before being processed, the second event is dropped.

4.5.1.3 L2 Cache and QDMA Queue Allocation

Similar to the EDMA channel controller, if the L2 controller queue allocation is exceeded on a single priority level then other requests cannot be submitted until the staling request is able to proceed.

The specific priority queue used for L2 cache requests is programmable using the CCFG.PRI register. Also, the queue allocation for each priority level is programmable via the L2ALLOC3, L2ALLOC2, L2ALLOC1, and L2ALLOC0 registers. The allocation programmed for the queue used for L2 cache requests is shared between L2 cache requests and QDMA requests. It is assured that the L2 controller will never submit more than six outstanding transfer requests per priority level at a given point in time. Therefore, the L2ALLOC*n* value (where *n* = CCFG.PRI value) should be set such that the total value is 6 plus the desired number of QDMAs outstanding for that priority level. If QDMAs are not submitted on that priority level, then a value of 6 will assure the allocation is never exceeded.

For priority levels other than the one used for cache requests, QDMA requests are the only request type for L2 transfer request submissions. The queue allocation for QDMA requests is limited to the value programmed into the L2ALLOC*n* registers (where *n*! = CCFG.PRI). If back-to-back QDMAs are submitted such that the QDMAs allocation is exceeded, this will stall all other L2 transfer requests (including QDMAs on other priority levels, and all L2 cache requests). This has a potential to stall the entire CPU since the L1D and L2 controllers can be stalled waiting for the QDMA request to be submitted.

In order to avoid this situation, it is recommended that the number of outstanding QDMA requests on a given priority level never reach the value programmed in the L2ALLOC*n* register. This can be done via inherent knowledge of the data flow of the system or can be manually tracked via software by using transfer completion code interrupts to track completion of QDMA requests.

4.5.1.4 Peripheral Queue Allocation

Peripheral requestors priority level and queue allocation are programmable via the TRCTL register in the given peripheral. Refer to peripheral-specific reference guide and device-specific data manual for details. In general, you should only modify the priority level used by the peripheral and not modify the programmable queue allocation field.

There are no interdependencies for peripheral queue allocation since a given peripheral always issues transfer requests on a single priority level. If a given request is stalled due to exceeding its allocation, there are no other concurrent requests to consider since all peripheral requests naturally occur in order.

4.5.2 Modifying Default Queue Usage and Queue Allocation

Queue usage and queue allocation for all requestors should not be modified dynamically during device operation. Allocation and priority usage should be set during system initialization.

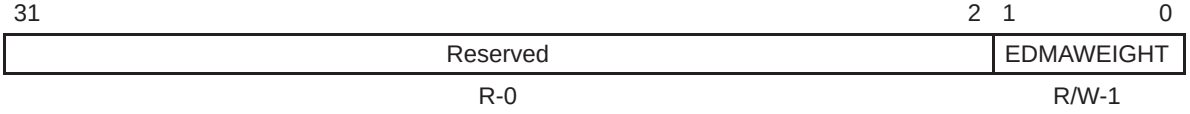
You must assure that the sum of all requestor allocations on a single priority level not exceed the queue depth (16 entries). For example, it is illegal to program HPI allocation, EDMA allocation, and L2 allocation all to a value of 7, since the sum of the three (7 + 7 + 7 = 21) exceeds the depth of the queue, which is 16. It is legal to use values of 7 + 7 + 2 = 16.

4.6 EDMA Access Into L2 Control

The C64x DSP incorporates an L2 EDMA access control register (EDMAWEIGHT), located in the L2 cache register memory map, that controls the relative priority weighting of EDMA versus L1D access into L2. EDMAWEIGHT gives EDMA accesses a temporary boost in priority by limiting the amount of time L1D blocks EDMA access to L2. This priority boost only applies when competing with write data from the CPU that misses in L1D, but hits in L2 cache or L2 SRAM. Normal line allocate and eviction operation result in gaps in the L1D-to-L2 access pattern that can be used by EDMA accesses. EDMAWEIGHT lets you control how often this priority boost is given. When the EDMA priority is raised, it is allowed to complete one access before priority is returned to the CPU data. For reference, the EDMAWEIGHT is shown in Figure 4−2 and described in Table 4−5.

If L1D blocks L2 for *n* consecutive cycles, then EDMA is given priority for a single cycle.

Figure 4−2. L2 EDMA Access Control Register (EDMAWEIGHT)



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

Table 4−5. L2 EDMA Access Control Register (EDMAWEIGHT) Field Descriptions

Bit	Field	Value	Description
31−2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1−0	EDMAWEIGHT	0−3h	EDMA weight limits the amount of time L1D blocks EDMA access to L2.
		0	L1D access always higher priority than EDMA access to L2. EDMA never receives priority.
		1h	EDMA receives priority after 16 L1D priority cycles.
		2h	EDMA receives priority after 4 L1D priority cycles.
		3h	EDMA receives priority after 1 L1D priority cycle.

4.7 EDMA Control Registers

Each of the 64 channels in the EDMA has a specific synchronization event associated with it. These events trigger the data transfer associated with that channel. The list of control registers that perform various processing of events is shown in Table 4−6. These synchronization events are discussed in detail in section 1.5. See the device-specific data manual for the memory address of these registers and the event-to-channel mapping.

Table 4−6. EDMA Control Registers—C64x DSP

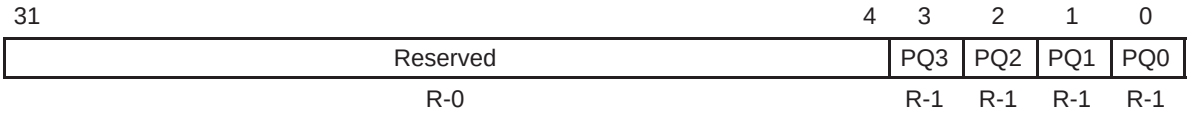
Acronym	Register Name	Section
PQSR	EDMA priority queue status register	4.7.1
PQAR	EDMA priority queue allocation registers	4.7.2
CIPRL	EDMA channel interrupt pending low register	4.7.3
CIPRH	EDMA channel interrupt pending high register	4.7.3
CIERL	EDMA channel interrupt enable low register	4.7.4
CIERH	EDMA channel interrupt enable high register	4.7.4
CCERL	EDMA channel chain enable low register	4.7.5
CCERH	EDMA channel chain enable high register	4.7.5
ERL	EDMA event low register	4.7.6
ERH	EDMA event high register	4.7.6
EERL	EDMA event enable low register	4.7.7
EERH	EDMA event enable high register	4.7.7
ECRL	EDMA event clear low register	4.7.8
ECRH	EDMA event clear high register	4.7.8
ESRL	EDMA event set low register	4.7.9
ESRH	EDMA event set high register	4.7.9
EPRL	EDMA event polarity low register	4.7.10
EPRH	EDMA event polarity high register	4.7.10

4.7.1 Priority Queue Status Register (PQSR)

The priority queue status register (PQSR) indicates whether the transfer controller is empty on each priority level. The PQSR is shown in Figure 4−3 and described in Table 4−7. The priority queue status (PQ) bit provides the status of the queues as well as any active transfers. When the PQ bits are set to 1111b, there are no requests pending in the respective priority level queues and no transfer is in progress. For example, if bit 0 (PQ0) is set to 1, all L2 requests for data movement have been completed and there are no requests pending in the priority level 0 queue.

The PQ bits are mainly used for emulation or debugging and typically should not be used by an application.

Figure 4−3. Priority Queue Status Register (PQSR)



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

Table 4−7. Priority Queue Status Register (PQSR) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−4	Reserved	−	0	Reserved. You should always write 0 to this field.
3−0	PQ	OF(*value*)	0−Fh	Priority queue status. A 1 in the PQ bit indicates that there are no requests pending in the respective priority level queue.
		DEFAULT	Fh	There are no requests pending in the priority level queues.

† For CSL implementation, use the notation EDMA_PQSR_PQ_*symval.*

4.7.2 Priority Queue Allocation Registers (PQAR0−3)

The C64x DSP has four transfer request queues: Q0, Q1, Q2, and Q3. The different priority level transfer requests (PRI field in the EDMA channel options parameter) are sorted into Q0, Q1, Q2, and Q3. The queue length available to EDMA requests is programmable using the priority queue allocation registers (PQAR*n*). The PQAR*n* is shown in Figure 4−4 and described in Table 4−8.

Figure 4−4. Priority Queue Allocation Register (PQAR)

31	4	3	0
Reserved		PQA†	
R-0		R/W-0 R/W-0 R/W-1 R/W-0	

Legend: R = Read only; R/W = Read/Write; -*n* = value after reset
† For PQAR0 and PQAR2, the default value is 0010b; for PQAR1 and PQAR3, the default value is 0110b.

Table 4−8. Priority Queue Allocation Register (PQAR) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−4	Reserved	−	0	Reserved. You should always write 0 to this field.
3−0	PQA	OF(*value*)	0−Fh	Priority queue allocation bits determine the queue length available to EDMA requests.
		DEFAULT	2h	For PQAR0 and PQAR2: Queue length of 2 is available to EDMA requests.
		DEFAULT	6h	For PQAR1 and PQAR3: Queue length of 6 is available to EDMA requests.

† For CSL implementation, use the notation EDMA_PQAR0_PQA_*symval*, EDMA_PQAR1_PQA_*symval*, EDMA_PQAR2_PQA_*symval*, and EDMA_PQAR3_PQA_*symval.*

4.7.3 EDMA Channel Interrupt Pending Registers (CIPRL, CIPRH)

The EDMA channel interrupt pending registers (CIPRL and CIPRH) are shown in Figure 4−5 and Figure 4−6 and described in Table 4−9 and Table 4−10.

4.7.3.1 EDMA Channel Interrupt Pending Low Register (CIPRL)

Figure 4−5. EDMA Channel Interrupt Pending Low Register (CIPRL)

31	30	29	28	27	26	25	24
CIP31	CIP30	CIP29	CIP28	CIP27	CIP26	CIP25	CIP24
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0
23	22	21	20	19	18	17	16
CIP23	CIP22	CIP21	CIP20	CIP19	CIP18	CIP17	CIP16
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0
15	14	13	12	11	10	9	8
CIP15	CIP14	CIP13	CIP12	CIP11	CIP10	CIP9	CIP8
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0
7	6	5	4	3	2	1	0
CIP7	CIP6	CIP5	CIP4	CIP3	CIP2	CIP1	CIP0
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

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

Table 4−9. EDMA Channel Interrupt Pending Low Register (CIPRL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	CIP	OF(*value*)	0−FFFF FFFFh	Channel 0−31 interrupt pending. When the TCINT or ATCINT bit in the channel options parameter (OPT) is set to 1 for an EDMA channel and a specific transfer complete code (TCC) or alternate transfer complete code (ATCC) is provided by the EDMA transfer controller, the EDMA channel controller sets a bit in the CIP field.
		DEFAULT	0	EDMA channel interrupt is not pending.
		−	1	EDMA channel interrupt is pending.

† For CSL implementation, use the notation EDMA_CIPRL_CIP_*symval.*

4.7.3.2 EDMA Channel Interrupt Pending High Register (CIPRH)

Figure 4−6. EDMA Channel Interrupt Pending High Register (CIPRH)

31	30	29	28	27	26	25	24
CIP63	CIP62	CIP61	CIP60	CIP59	CIP58	CIP57	CIP56
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0
23	22	21	20	19	18	17	16
CIP55	CIP54	CIP53	CIP52	CIP51	CIP50	CIP49	CIP48
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0
15	14	13	12	11	10	9	8
CIP47	CIP46	CIP45	CIP44	CIP43	CIP42	CIP41	CIP40
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0
7	6	5	4	3	2	1	0
CIP39	CIP38	CIP37	CIP36	CIP35	CIP34	CIP33	CIP32
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

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

Table 4−10. EDMA Channel Interrupt Pending High Register (CIPRH) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	CIP	OF(*value*)	0−FFFF FFFFh	Channel 32−63 interrupt pending. When the TCINT or ATCINT bit in the channel options parameter (OPT) is set to 1 for an EDMA channel and a specific transfer complete code (TCC) or alternate transfer complete code (ATCC) is provided by the EDMA transfer controller, the EDMA channel controller sets a bit in the CIP field.
		DEFAULT	0	EDMA channel interrupt is not pending.
		−	1	EDMA channel interrupt is pending.

† For CSL implementation, use the notation EDMA_CIPRH_CIP_*symval.*

4.7.4 EDMA Channel Interrupt Enable Registers (CIERL, CIERH)

The EDMA channel interrupt enable registers (CIERL and CIERH) are shown in Figure 4−7 and Figure 4−8 and described in Table 4−11 and Table 4−12.

4.7.4.1 EDMA Channel Interrupt Enable Low Register (CIERL)

Figure 4−7. EDMA Channel Interrupt Enable Low Register (CIERL)

31	30	29	28	27	26	25	24
CIE31	CIE30	CIE29	CIE28	CIE27	CIE26	CIE25	CIE24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
CIE23	CIE22	CIE21	CIE20	CIE19	CIE18	CIE17	CIE16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
CIE15	CIE14	CIE13	CIE12	CIE11	CIE10	CIE9	CIE8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
CIE7	CIE6	CIE5	CIE4	CIE3	CIE2	CIE1	CIE0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 4−11. EDMA Channel Interrupt Enable Low Register (CIERL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	CIE	OF(*value*)	0−FFFF FFFFh	Channel 0−31 interrupt enable. A 32-bit unsigned value used to disable or enable an interrupt for an EDMA channel.
		DEFAULT	0	EDMA channel interrupt is not enabled.
		−	1	EDMA channel interrupt is enabled.

† For CSL implementation, use the notation EDMA_CIERL_CIE_*symval.*

4.7.4.2 EDMA Channel Interrupt Enable High Register (CIERH)

Figure 4−8. EDMA Channel Interrupt Enable High Register (CIERH)

31	30	29	28	27	26	25	24
CIE63	CIE62	CIE61	CIE60	CIE59	CIE58	CIE57	CIE56
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
CIE55	CIE54	CIE53	CIE52	CIE51	CIE50	CIE49	CIE48
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
CIE47	CIE46	CIE45	CIE44	CIE43	CIE42	CIE41	CIE40
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
CIE39	CIE38	CIE37	CIE36	CIE35	CIE34	CIE33	CIE32
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 4−12. EDMA Channel Interrupt Enable High Register (CIERH) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	CIE	OF(*value*)	0−FFFF FFFFh	Channel 32−63 interrupt enable. A 32-bit unsigned value used to disable or enable an interrupt for an EDMA channel.
		DEFAULT	0	EDMA channel interrupt is not enabled.
		–	1	EDMA channel interrupt is enabled.

† For CSL implementation, use the notation EDMA_CIERH_CIE_*symval.*

4.7.5 EDMA Channel Chain Enable Registers (CCERL, CCERH)

The EDMA channel chain enable registers (CCERL and CCERH) are shown in Figure 4−9 and Figure 4−10 and described in Table 4−13 and Table 4−14.

4.7.5.1 EDMA Channel Chain Enable Low Register (CCERL)

Figure 4−9. EDMA Channel Chain Enable Low Register (CCERL)

31	30	29	28	27	26	25	24
CCE31	CCE30	CCE29	CCE28	CCE27	CCE26	CCE25	CCE24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

23	22	21	20	19	18	17	16
CCE23	CCE22	CCE21	CCE20	CCE19	CCE18	CCE17	CCE16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	9	8
CCE15	CCE14	CCE13	CCE12	CCE11	CCE10	CCE9	CCE8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
CCE7	CCE6	CCE5	CCE4	CCE3	CCE2	CCE1	CCE0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 4−13. EDMA Channel Chain Enable Low Register (CCERL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	CCE	OF(*value*)	0−FFFF FFFFh	Channel 0−31 chain enable. To enable the EDMA controller to chain channels by way of a single event, set the TCINT or ATCINT bit in the channel options parameter (OPT) to 1. Additionally, set the relevant bit in the CCE field to trigger off the next channel transfer specified by the transfer complete code (TCC) or alternate transfer complete code (ATCC).
		DEFAULT	0	EDMA channel chain is not enabled.
		−	1	EDMA channel chain is enabled.

† For CSL implementation, use the notation EDMA_CCERL_CCE_*symval.*

4.7.5.2 EDMA Channel Chain Enable High Register (CCERH)

Figure 4−10. EDMA Channel Chain Enable High Register (CCERH)

31	30	29	28	27	26	25	24
CCE63	CCE62	CCE61	CCE60	CCE59	CCE58	CCE57	CCE56
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
CCE55	CCE54	CCE53	CCE52	CCE51	CCE50	CCE49	CCE48
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
CCE47	CCE46	CCE45	CCE44	CCE43	CCE42	CCE41	CCE40
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
CCE39	CCE38	CCE37	CCE36	CCE35	CCE34	CCE33	CCE32
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 4−14. EDMA Channel Chain Enable High Register (CCERH) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	CCE	OF(*value*)	0−FFFF FFFFh	Channel 32−63 chain enable. To enable the EDMA controller to chain channels by way of a single event, set the TCINT or ATCINT bit in the channel options parameter (OPT) to 1. Additionally, set the relevant bit in the CCE field to trigger off the next channel transfer specified by the transfer complete code (TCC) or alternate transfer complete code (ATCC).
		DEFAULT	0	EDMA channel chain is not enabled.
		−	1	EDMA channel chain is enabled.

† For CSL implementation, use the notation EDMA_CCERH_CCE_*symval.*

4.7.6 EDMA Event Registers (ERL, ERH)

All events are captured in the event low register (ERL) and event high register (ERH) for the 64 channels, even when the events are disabled. The ERL is shown in Figure 4−11 and described in Table 4−15, and the ERH is shown in Figure 4−12 and described in Table 4−16. Section 1.5 describes the type of synchronization events and the EDMA channels associated with each of them.

4.7.6.1 EDMA Event Low Register (ERL)

Figure 4−11. EDMA Event Low Register (ERL)

31	30	29	28	27	26	25	24
EVT31	EVT30	EVT29	EVT28	EVT27	EVT26	EVT25	EVT24
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
23	22	21	20	19	18	17	16
EVT23	EVT22	EVT21	EVT20	EVT19	EVT18	EVT17	EVT16
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8
EVT15	EVT14	EVT13	EVT12	EVT11	EVT10	EVT9	EVT8
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
EVT7	EVT6	EVT5	EVT4	EVT3	EVT2	EVT1	EVT0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

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

Table 4−15. EDMA Event Low Register (ERL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	EVT	OF(*value*)	0−FFFF FFFFh	Event 0−31. Events 0−31 captured by the EDMA are latched in ERL, even if that event is disabled (EERL = 0).
		DEFAULT	0	EDMA event is not asserted.
		−	1	EDMA event is asserted.

† For CSL implementation, use the notation EDMA_ERL_EVT_*symval.*

4.7.6.2 EDMA Event High Register (ERH)

Figure 4−12. EDMA Event High Register (ERH)

31	30	29	28	27	26	25	24
EVT63	EVT62	EVT61	EVT60	EVT59	EVT58	EVT57	EVT56
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
23	22	21	20	19	18	17	16
EVT55	EVT54	EVT53	EVT52	EVT51	EVT50	EVT49	EVT48
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8
EVT47	EVT46	EVT45	EVT44	EVT43	EVT42	EVT41	EVT40
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
EVT39	EVT38	EVT37	EVT36	EVT35	EVT34	EVT33	EVT32
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

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

Table 4−16. EDMA Event High Register (ERH) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	EVT	OF(*value*)	0−FFFF FFFFh	Event 32−63. Events 32−63 captured by the EDMA are latched in ERH, even if that event is disabled (EERH = 0).
		DEFAULT	0	EDMA event is not asserted.
		–	1	EDMA event is asserted.

† For CSL implementation, use the notation EDMA_ERH_EVT_*symval.*

4.7.7 EDMA Event Enable Registers (EERL, EERH)

Each event in the event registers (ERL and ERH) can be enabled or disabled using the event enable registers (EERL and EERH). Any of the event bits in EER can be set to 1 to enable that corresponding event or can be cleared to 0 to disable that corresponding event. The EERL is shown in Figure 4−13 and described in Table 4−17, and the EERH is shown in Figure 4−14 and described in Table 4−18.

The event registers latch all events that are captured by the EDMA, even if that event is disabled. This is analogous to an interrupt enable and interrupt pending register for interrupt processing, thus ensuring that the EDMA does not drop any events. Reenabling an event with a pending event signaled in the event registers forces the EDMA controller to process that event according to its priority.

4.7.7.1 EDMA Event Enable Low Register (EERL)

Figure 4−13. EDMA Event Enable Low Register (EERL)

31	30	29	28	27	26	25	24
EE31	EE30	EE29	EE28	EE27	EE26	EE25	EE24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

23	22	21	20	19	18	17	16
EE23	EE22	EE21	EE20	EE19	EE18	EE17	EE16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	9	8
EE15	EE14	EE13	EE12	EE11	EE10	EE9	EE8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
EE7	EE6	EE5	EE4	EE3	EE2	EE1	EE0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 4−17. EDMA Event Enable Low Register (EERL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	EE	OF(*value*)	0−FFFF FFFFh	Event 0−31 enable. Used to enable or disable an event.
		DEFAULT	0	EDMA event is not enabled.
		−	1	EDMA event is enabled.

† For CSL implementation, use the notation EDMA_EERL_EE_*symval.*

4.7.7.2 EDMA Event Enable High Register (EERH)

Figure 4−14. EDMA Event Enable High Register (EERH)

31	30	29	28	27	26	25	24
EE63	EE62	EE61	EE60	EE59	EE58	EE57	EE56
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
EE55	EE54	EE53	EE52	EE51	EE50	EE49	EE48
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
EE47	EE46	EE45	EE44	EE43	EE42	EE41	EE40
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
EE39	EE38	EE37	EE36	EE35	EE34	EE33	EE32
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 4−18. EDMA Event Enable High Register (EERH) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	EE	OF(*value*)	0−FFFF FFFFh	Event 32−63 enable. Used to enable or disable an event.
		DEFAULT	0	EDMA event is not enabled.
		−	1	EDMA event is enabled.

† For CSL implementation, use the notation EDMA_EERH_EE_*symval.*

4.7.8 Event Clear Registers (ECRL, ECRH)

Once an event has been posted in the event registers (ERL and ERH), the event can be cleared in two ways. If the event is enabled in the event enable registers (EERL and EERH) and the EDMA submits a transfer request for that event, it clears the corresponding event bit in the event register. Alternatively, if the event is disabled in the event enable register, the CPU can clear the event by way of the event clear low register (ECRL) or event clear high register (ECRH). The ECRL is shown in Figure 4−15 and described in Table 4−19, and the ECRH is shown in Figure 4−16 and described in Table 4−20.

Writing a 1 to any of the bits clears the corresponding event; writing a 0 has no effect. Once an event bit is set in the event register, it remains set until the EDMA submits a transfer request for that event or the CPU clears the event by setting the corresponding bit in ECRL or ECRH.

4.7.8.1 EDMA Event Clear Low Register (ECRL)

Figure 4−15. EDMA Event Clear Low Register (ECRL)

31	30	29	28	27	26	25	24
EC31	EC30	EC29	EC28	EC27	EC26	EC25	EC24
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0
23	22	21	20	19	18	17	16
EC23	EC22	EC21	EC20	EC19	EC18	EC17	EC16
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0
15	14	13	12	11	10	9	8
EC15	EC14	EC13	EC12	EC11	EC10	EC9	EC8
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0
7	6	5	4	3	2	1	0
EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0

Legend: W1C = Write 1 to clear (writing 0 has no effect); − *n* = value after reset

Table 4−19. EDMA Event Clear Low Register (ECRL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	EC	OF(*value*)	0−FFFF FFFFh	Event 0−31 clear. Any of the event bits can be set to 1 to clear that event; a write of 0 has no effect.
		DEFAULT	0	No effect.
		−	1	EDMA event is cleared.

† For CSL implementation, use the notation EDMA_ECRL_EC_*symval.*

4.7.8.2 EDMA Event Clear High Register (ECRH)

Figure 4−16. EDMA Event Clear High Register (ECRH)

31	30	29	28	27	26	25	24
EC63	EC62	EC61	EC60	EC59	EC58	EC57	EC56
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0
23	22	21	20	19	18	17	16
EC55	EC54	EC53	EC52	EC51	EC50	EC49	EC48
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0
15	14	13	12	11	10	9	8
EC47	EC46	EC45	EC44	EC43	EC42	EC41	EC40
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0
7	6	5	4	3	2	1	0
EC39	EC38	EC37	EC36	EC35	EC34	EC33	EC32
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0

Legend: W1C = Write 1 to clear (writing 0 has no effect); − *n* = value after reset

Table 4−20. EDMA Event Clear High Register (ECRH) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	EC	OF(*value*)	0−FFFF FFFFh	Event 32−63 clear. Any of the event bits can be set to 1 to clear that event; a write of 0 has no effect.
		DEFAULT	0	No effect.
		−	1	EDMA event is cleared.

† For CSL implementation, use the notation EDMA_ECRH_EC_*symval.*

4.7.9 Event Set Registers (ESRL, ESRH)

The CPU can set events using the event set registers (ESRL and ESRH). The ESRL is shown in Figure 4−17 and described in Table 4−21, and the ESRH is shown in Figure 4−18 and described in Table 4−22. Writing a 1 to one of the event bits causes a transfer request to be submitted. The event does not have to be enabled. This provides a debugging tool and also allows the CPU to submit EDMA requests in the system. Note that CPU-initiated EDMA transfers are considered CPU-synchronized transfers. In other words, an EDMA transfer occurs when the corresponding ESRL or ESRH bit is set and is not triggered by the associated event. See section 1.16 for a description of the quick DMA (QDMA), an alternative way to perform CPU-initiated EDMA transfers.

4.7.9.1 EDMA Event Set Low Register (ESRL)

Figure 4−17. EDMA Event Set Low Register (ESRL)

31	30	29	28	27	26	25	24
ES31	ES30	ES29	ES28	ES27	ES26	ES25	ES24
W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0

23	22	21	20	19	18	17	16
ES23	ES22	ES21	ES20	ES19	ES18	ES17	ES16
W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0

15	14	13	12	11	10	9	8
ES15	ES14	ES13	ES12	ES11	ES10	ES9	ES8
W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0

7	6	5	4	3	2	1	0
ES7	ES6	ES5	ES4	ES3	ES2	ES1	ES0
W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0

Legend: W1S = Write 1 to set (writing 0 has no effect); − *n* = value after reset

Table 4−21. EDMA Event Set Low Register (ESRL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	ES	OF(*value*)	0−FFFF FFFFh	Event 0−31 set. Any of the event bits can be set to 1 to set the corresponding bit in the event low register (ERL); a write of 0 has no effect.
		DEFAULT	0	No effect.
		−	1	EDMA event is set.

† For CSL implementation, use the notation EDMA_ESRL_ES_*symval.*

4.7.9.2 EDMA Event Set High Register (ESRH)

Figure 4−18. EDMA Event Set High Register (ESRH)

31	30	29	28	27	26	25	24
ES63	ES62	ES61	ES60	ES59	ES58	ES57	ES56
W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0
23	22	21	20	19	18	17	16
ES55	ES54	ES53	ES52	ES51	ES50	ES49	ES48
W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0
15	14	13	12	11	10	9	8
ES47	ES46	ES45	ES44	ES43	ES42	ES41	ES40
W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0
7	6	5	4	3	2	1	0
ES39	ES38	ES37	ES36	ES35	ES34	ES33	ES32
W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0	W1S-0

Legend: W1S = Write 1 to set (writing 0 has no effect); − *n* = value after reset

Table 4−22. EDMA Event Set High Register (ESRH) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	ES	OF(*value*)	0−FFFF FFFFh	Event 32−63 set. Any of the event bits can be set to 1 to set the corresponding bit in the event high register (ERH); a write of 0 has no effect.
		DEFAULT	0	No effect.
		−	1	EDMA event is set.

† For CSL implementation, use the notation EDMA_ESRH_ES_*symval.*

4.7.10 Event Polarity Registers (EPRL, EPRH)

An event is signaled to the EDMA controller by a positive-edge triggering (low-to-high transition) on one of its event inputs. The event polarity can be changed to a falling-edge triggering (high-to-low transition) by setting the corresponding bit in the event polarity low register (EPRL) or event polarity high register (EPRH). In general, the event polarity should only be inverted for externally-provided events. The EPRL is shown in Figure 4−19 and described in Table 4−23, and the EPRH is shown in Figure 4−20 and described in Table 4−24.

4.7.10.1 EDMA Event Polarity Low Register (EPRL)

Figure 4−19. EDMA Event Polarity Low Register (EPRL)

31	30	29	28	27	26	25	24
EP31	EP30	EP29	EP28	EP27	EP26	EP25	EP24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
EP23	EP22	EP21	EP20	EP19	EP18	EP17	EP16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 4−23. EDMA Event Polarity Low Register (EPRL) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	EP	OF(*value*)	0−FFFF FFFFh	Event 0−31 polarity. A 32-bit unsigned value used to select a rising edge or falling edge to determine when an event is triggered on its input.
		DEFAULT	0	EDMA event is triggered on a rising edge on its input.
		−	1	EDMA event is triggered on a falling edge on its input.

† For CSL implementation, use the notation EDMA_EPRL_EP_*symval.*

4.7.10.2 EDMA Event Polarity High Register (EPRH)

Figure 4−20. EDMA Event Polarity High Register (EPRH)

31	30	29	28	27	26	25	24
EP63	EP62	EP61	EP60	EP59	EP58	EP57	EP56
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
EP55	EP54	EP53	EP52	EP51	EP50	EP49	EP48
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
EP47	EP46	EP45	EP44	EP43	EP42	EP41	EP40
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
EP39	EP38	EP37	EP36	EP35	EP34	EP33	EP32
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

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

Table 4−24. EDMA Event Polarity High Register (EPRH) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	EP	OF(*value*)	0−FFFF FFFFh	Event 32−63 polarity. A 32-bit unsigned value used to select a rising edge or falling edge to determine when an event is triggered on its input.
		DEFAULT	0	EDMA event is triggered on a rising edge on its input.
		–	1	EDMA event is triggered on a falling edge on its input.

† For CSL implementation, use the notation EDMA_EPRH_EP_*symval.*

4.8 EDMA Channel Parameter Entries

See section 1.3 for an overview of the parameter RAM (PaRAM). Each parameter set of an EDMA channel is organized into six 32-bit words or 24 bytes as listed in Table 4−25. See Table 4−1 (page 4-3) for the memory address of these registers.

Table 4−25. EDMA Parameter Entries—C64x DSP

Acronym	Parameter Name	Section
OPT	EDMA channel options parameter	4.8.1
SRC	EDMA channel source address parameter	4.8.2
CNT	EDMA channel transfer count parameter	4.8.3
DST	EDMA channel destination address parameter	4.8.4
IDX	EDMA channel index parameter	4.8.5
RLD	EDMA channel count reload/link address parameter	4.8.6

4.8.1 EDMA Channel Options Parameter (OPT)

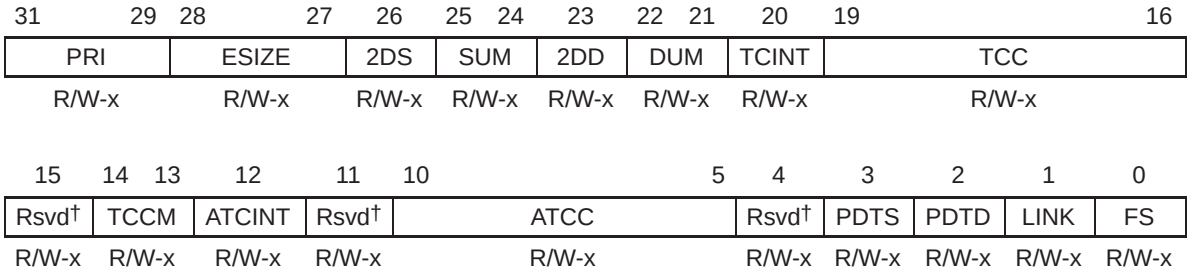
The EDMA channel options parameter (OPT) in the EDMA parameter entries is shown in Figure 4−21 and described in Table 4−26.

PDT transfers are classified in terms of the memory on the EMIF. A PDT write is a transfer from a peripheral to memory (memory is physically written). To enable a PDT write from an external peripheral source to an external memory destination, set the PDTD bit in the channel options parameter (OPT) to 1.

A PDT read is a transfer from memory to a peripheral (memory is physically read). To enable a PDT read from an external memory source to an external peripheral destination, set the PDTS bit in OPT to 1.

PDT writes and PDT reads are mutually exclusive. In other words, PDTS and PDTD cannot both be set to 1.

Figure 4−21. EDMA Channel Options Parameter (OPT)



† Always write 0 to the reserved bits.
Legend: R/W = Read/Write; -x = value is indeterminate after reset

Table 4−26. EDMA Channel Options Parameter (OPT) Field Descriptions

Bit	***field***†	***symval***†	Value	Description
31−29	PRI	OF(*value*)	0−7h	Priority levels for EDMA events.
		DEFAULT URGENT	0	Urgent priority EDMA transfer.
		HIGH	1h	High priority EDMA transfer.
		MEDIUM	2h	Medium priority EDMA transfer.
		LOW	3h	Low priority EDMA transfer.
		−	4h−7h	Reserved.
28−27	ESIZE	OF(*value*)	0−3h	Element size.
		DEFAULT 32BIT	0	32-bit word, or 64-bit doubleword (on certain C64x EDMA transfers only, see section 1.7).
		16BIT	1h	16-bit halfword.
		8BIT	2h	8-bit byte.
		−	3h	Reserved.
26	2DS	OF(*value*)		Source dimension.
		DEFAULT NO	0	1-dimensional source.
		YES	1	2-dimensional source.

† For CSL implementation, use the notation EDMA_OPT_*field_symval.*

Table 4−26. EDMA Channel Options Parameter (OPT) Field Descriptions (Continued)

Bit	*field*†	*symval*†	Value	Description
25−24	SUM	OF(*value*)	0−3h	Source address update mode.
		DEFAULT NONE	0	Fixed address mode. No source address modification.
		INC	1h	Source address increment depends on the 2DS and FS bits.
		DEC	2h	Source address decrement depends on the 2DS and FS bits.
		IDX	3h	Source address modified by the element index/frame index depending on the 2DS and FS bits.
23	2DD	OF(*value*)		Destination dimension.
		DEFAULT NO	0	1-dimensional destination.
		YES	1	2-dimensional destination.
22−21	DUM	OF(*value*)	0−3h	Destination address update mode.
		DEFAULT NONE	0	Fixed address mode. No destination address modification.
		INC	1h	Destination address increment depends on the 2DD and FS bits.
		DEC	2h	Destination address decrement depends on the 2DD and FS bits.
		IDX	3h	Destination address modified by the element index/frame index depending on the 2DD and FS bits.
20	TCINT	OF(*value*)		Transfer complete interrupt.
		DEFAULT NO	0	Transfer complete indication is disabled. The EDMA channel interrupt pending register (CIPRL or CIPRH) bits are not set upon completion of a transfer.
		YES	1	Transfer complete indication is enabled. The EDMA channel interrupt pending register (CIPRL or CIPRH) bit is set on a channel transfer completion. The bit (position) set in CIPRL or CIPRH is the TCC value specified. This bit can be used for chaining and interrupt generation.

† For CSL implementation, use the notation EDMA_OPT_*field_symval.*

Table 4−26. EDMA Channel Options Parameter (OPT) Field Descriptions (Continued)

Bit	*field*†	*symval*†	Value	Description
19−16	TCC	OF(*value*)	0−Fh	Transfer complete code. This 4-bit value is used to set the bit in the EDMA channel interrupt pending register (CIPR[TCC] bit) provided. TCC works in conjunction with the TCCM bits to provide a 6-bit transfer complete code. This bit can be used for chaining and interrupt generation.
		DEFAULT	0	
15	Reserved	−	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. If writing to this field, always write a 0.
14−13	TCCM	OF(*value*)	0−3h	Transfer complete code most-significant bits. This 2-bit value works in conjunction with the TCC bits to provide a 6-bit transfer complete code. The 6-bit code is used to set the relevant bit in the EDMA channel interrupt pending register (CIPRL or CIPRH) provided TCINT = 1, when the current set is exhausted.
		DEFAULT	0	
12	ATCINT	OF(*value*)		Alternate transfer complete interrupt.
		DEFAULT NO	0	Alternate transfer complete indication is disabled. The EDMA channel interrupt pending register (CIPRL or CIPRH) bits are not set upon completion of intermediate transfers in a block.
		YES	1	Alternate transfer complete indication is enabled. The EDMA channel interrupt pending register (CIPRL or CIPRH) bit is set upon completion of intermediate transfers in a block. The bit (position) set in CIPRL or CIPRH is the ATCC value specified.
11	Reserved	−	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. If writing to this field, always write a 0.
10−5	ATCC	OF(*value*)	0−3Fh	Alternate transfer complete code. This 6-bit value is used to set the bit in the EDMA channel interrupt pending register (CIPRL or CIPRH) (CIP[ATCC] bit) provided ATCINT = 1, upon completion of an intermediate transfer in a block. This bit can be used for chaining and interrupt generation.
		DEFAULT	0	
4	Reserved	−	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect. If writing to this field, always write a 0.

† For CSL implementation, use the notation EDMA_OPT_*field_symval.*

Table 4−26. EDMA Channel Options Parameter (OPT) Field Descriptions (Continued)

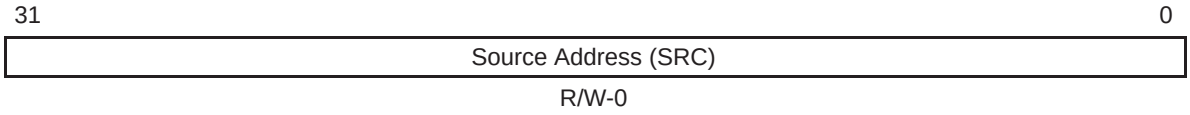
Bit	*field*†	*symval*†	Value	Description
3	PDTS	OF(*value*)		Peripheral device transfer (PDT) mode for source.
		DEFAULT DISABLE	0	PDT read is disabled.
		ENABLE	1	PDT read is enabled.
2	PDTD	OF(*value*)		Peripheral device transfer (PDT) mode for destination.
		DEFAULT DISABLE	0	PDT write is disabled.
		ENABLE	1	PDT write is enabled.
1	LINK	OF(*value*)		Linking of event parameters enable.
		DEFAULT NO	0	Linking of event parameters is disabled. Entry is not reloaded.
		YES	1	Linking of event parameters is enabled. After the current set is exhausted, the channel entry is reloaded with the parameter set specified by the link address.
0	FS	OF(*value*)		Frame synchronization.
		DEFAULT NO	0	Channel is element/array synchronized.
		YES	1	Channel is frame synchronized. The relevant event for a given EDMA channel is used to synchronize a frame.

† For CSL implementation, use the notation EDMA_OPT_*field_symval.*

4.8.2 EDMA Channel Source Address Parameter (SRC)

The EDMA channel source address parameter (SRC) in the EDMA parameter entries specifies the starting byte address of the source. The SRC is shown in Figure 4−22 and described in Table 4−27. Use the SUM bits in the EDMA channel options parameter (OPT) to modify the source address. See section 1.9.2 for details. The source address must be aligned to the value specified by ESIZE, refer to section 1.7.

Figure 4−22. EDMA Channel Source Address Parameter (SRC)



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

Table 4−27. EDMA Channel Source Address Parameter (SRC) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	SRC	OF(*value*)	0−FFFF FFFFh	This 32-bit source address specifies the starting byte address of the source. The address is modified using the SUM bits in the EDMA channel options parameter (OPT).
		DEFAULT	0	

† For CSL implementation, use the notation EDMA_SRC_SRC_*symval.*

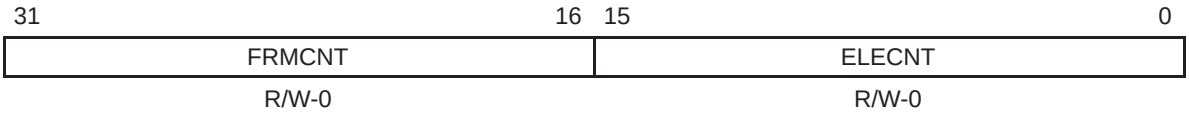
4.8.3 EDMA Channel Transfer Count Parameter (CNT)

The EDMA channel transfer count parameter (CNT) in the EDMA parameter entries specifies the frame/array count and element count. The CNT is shown in Figure 4−23 and described in Table 4−28.

The frame/array count (FRMCNT) is a 16-bit unsigned value plus 1 that specifies the number of frames in a 1D block or the number of arrays in a 2D block. Frame count applies to 1D transfers and array count applies to 2D transfers. Valid values for the frame/array count range between 0 and 65535; therefore, the maximum number of frames/arrays in a block is 65536. A frame/array count of 0 is actually one frame/array, and a frame/array count of 1 is two frames/arrays. See section 1.9.1 for details.

The element count (ELECNT) is a 16-bit unsigned value that specifies the number of elements in a frame (for 1D transfers) or in an array (for 2D transfers). Valid values for the element count range between 1 and 65535; therefore, the maximum number of elements in a frame is 65535. The EDMA performs no transfers if the element count is 0. See section 1.9.1 for details.

Figure 4−23. EDMA Channel Transfer Count Parameter (CNT)



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

Table 4−28. EDMA Channel Transfer Count Parameter (CNT) Field Descriptions

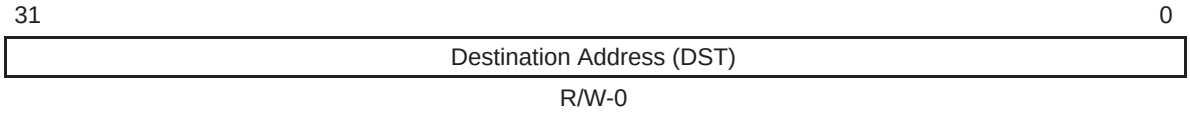
Bit	***field***†	***symval***†	Value	Description
31−16	FRMCNT	OF(*value*)	0−FFFFh	Frame/array count. A 16-bit unsigned value plus 1 that specifies the number of frames in a 1D block or number of arrays in a 2D block. Valid values for the frame/array count: 0−65535.
		DEFAULT	0	1 frame in a 1D block or 1 array in a 2D block.
15−0	ELECNT	OF(*value*)	0−FFFFh	Element count. A 16-bit unsigned value that specifies the number of elements in a frame (for 1D transfers) or an array (for 2D transfers). Valid values for the element count: 1−65535.
		DEFAULT	0	No transfer.

† For CSL implementation, use the notation EDMA_CNT_*field_symval.*

4.8.4 EDMA Channel Destination Address Parameter (DST)

The EDMA channel destination address parameter (DST) in the EDMA parameter entries specifies the starting byte address of the destination. The DST is shown in Figure 4−24 and described in Table 4−29. Use the DUM bits in the EDMA channel options parameter (OPT) to modify the destination address. See section 1.9.2 for details. The destination address must be aligned to the value specified by ESIZE, refer to section 1.7.

Figure 4−24. EDMA Channel Destination Address Parameter (DST)



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

Table 4−29. EDMA Channel Destination Address Parameter (DST) Field Descriptions

Bit	Field	*symval*†	Value	Description
31−0	DST	OF(*value*)	0−FFFF FFFFh	This 32-bit destination address specifies the starting byte address of the destination. The address is modified using the DUM bits in the EDMA channel options parameter (OPT).
		DEFAULT	0	

† For CSL implementation, use the notation EDMA_DST_DST_*symval.*

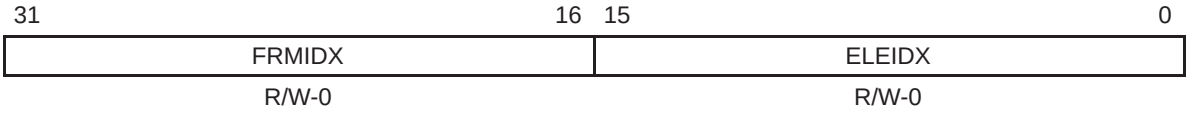
4.8.5 EDMA Channel Index Parameter (IDX)

The EDMA channel index parameter (IDX) in the EDMA parameter entries specifies the frame/array index and element index used for address modification. The EDMA is shown in Figure 4−25 and described in Table 4−30. The EDMA uses the indexes for address updates, depending on the type of transfer (1D or 2D) selected, and the FS, SUM, and DUM bits in the EDMA channel options parameter (OPT).

The frame/array index (FRMIDX) is a 16-bit signed value that specifies the address offset (in bytes) to the next frame/array in a 1D transfer or 2D transfer. Frame index applies to 1D transfers and array index applies to 2D transfers. Valid values for the frame/array index range between –32768 to 32767.

The element index (ELEIDX) is a 16-bit signed value that specifies the address offset (in bytes) to the next element in a frame. The element index is used only for 1D transfers, because 2D transfers do not allow spacing between elements. Valid values for the element index range between –32768 to 32767.

Figure 4−25. EDMA Channel Index Parameter (IDX)



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

Table 4−30. EDMA Channel Index Parameter (IDX) Field Descriptions

Bit	***field***†	***symval***†	Value	Description
31−16	FRMIDX	OF(*value*)	0−FFFFh	Frame/array index. A 16-bit signed value that specifies the frame/array index used for an address offset to the next frame/array. Valid values for the frame/array index: –32768 to 32767.
		DEFAULT	0	No offset is used as the frame/array index for an address offset to the next frame/array.
15−0	ELEIDX	OF(*value*)	0−FFFFh	Element index. A 16-bit signed value that specifies the element index used for an address offset to the next element in a frame. Element index is used *only* for 1D transfers. Valid values for the element index: –32768 to 32767.
		DEFAULT	0	No offset is used as the element index for an address offset to the next element in a frame.

† For CSL implementation, use the notation EDMA_IDX_*field_symval.*

4.8.6 EDMA Channel Count Reload/Link Address Parameter (RLD)

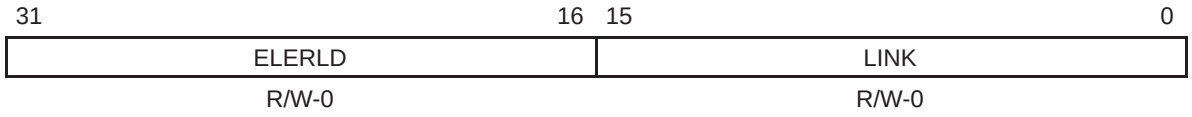
The EDMA channel count reload/link address parameter (RLD) in the EDMA parameter entries specifies the value used to reload the element count field and the link address. The RLD is shown in Figure 4−26 and described in Table 4−31.

The 16-bit unsigned element count reload (ELERLD) value reloads the element count (ELECNT) field in the EDMA channel transfer count parameter (CNT), once the last element in a frame is transferred. ELERLD is used only for a 1D element sync (FS = 0) transfer, since the EDMA has to keep track of the next element address using the element count. This is necessary for multi-frame EDMA transfers, where frame count value is greater than 0. See section 1.9.1 for more details.

The EDMA controller provides a mechanism to link EDMA transfers for auto-initialization. When LINK = 1 in the EDMA channel options parameter (OPT), the 16-bit link address (LINK) specifies the lower 16-bit address in the parameter RAM where the EDMA loads/reloads the parameters in preparation for the next event. Since the entire EDMA parameter RAM is located in the 01A0 xxxxh area, only the lower 16-bit address is required.

The reload parameters are specified in the address range starting at 01A0 0600h. You must ensure that the link address is on a 24-byte boundary, and that the operation is undefined if this rule is violated (see section 1.11). In addition to the reload parameter space, the entry of any unused EDMA channel can also be used for linking. The EDMA can always have up to 85 programmed entries, regardless of the number of channels actually used.

Figure 4−26. EDMA Channel Count Reload/Link Address Parameter (RLD)



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

Table 4−31. EDMA Channel Count Reload/Link Address Parameter (RLD) Field Descriptions

Bit	***field***†	***symval***†	Value	Description
31−16	ELERLD	OF(*value*)	0−FFFFh	Element count reload. A 16-bit unsigned value used to reload the element count field in the EDMA channel transfer count parameter (CNT) once the last element in a frame is transferred. This field is used only for a 1D element sync (FS = 0) transfer, since the EDMA has to keep track of the next element address using the element count. This is necessary for multi-frame EDMA transfers where frame count value is greater than 0.
		DEFAULT	0	0 is used to reload the element count field in the EDMA channel transfer count parameter (CNT) once the last element in a frame is transferred.
15−0	LINK	OF(*value*)	0−FFFFh	This 16-bit link address specifies the lower 16-bit address in the parameter RAM from which the EDMA loads/reloads the parameters of the next event in the chain.
		DEFAULT	0	The address 01A0 0000h in the parameter RAM is used to load/reload the parameters of the next event in the chain.

† For CSL implementation, use the notation EDMA_RLD_*field_symval.*

4.9 QDMA Registers

Since the QDMA is used for quick, one-time transfers it does not have the capability to reload a count or link. The count reload/link address register is therefore not available to the QDMA. The QDMA registers are not updated during or after a transfer by the hardware, they retain the submitted values. All QDMA transfers are submitted using frame synchronization (1D) or block synchronization (2D). See section 1.16 for more details.

The QDMA consists of two sets of memory-mapped registers (Figure 4−27), similar to an EDMA parameter entry. The first set shown in Figure 4−27(a) is a direct mapping of the five QDMA registers required to configure a transfer. There is no count reload, no link address, and the LINK field of the QDMA channel options register (QOPT) is reserved. Writing to the QDMA registers configures, but does not submit, a QDMA transfer request. Figure 4−27(b) shows the pseudo-registers for this set. Writing to any of the pseudo-registers submits a transfer request.

Although the QDMA mechanism does not support event linking, it supports completion interrupts, as well as QDMA transfer-complete chaining to EDMA channels. QDMA completion interrupts are enabled and set in the same way as EDMA completion interrupts; through the use of the TCINT and TCC bits in the QDMA channel options register (QOPT), and CIPR and CIER of the EDMA. QDMA transfer-complete chaining with EDMA events are enabled through setting the appropriate bits in QOPT and CCER of the EDMA channel controller. QDMA transfer requests have the same priority restrictions as the EDMA. See section 4.5 for details.

Access to each register is limited to 32-bits only. Halfword and byte writes result in undefined behavior.

Figure 4−27. QDMA Registers

(a) QDMA registers

Address			QDMA register
0200 0000h	QDMA Channel Options		QOPT
0200 0004h	QDMA Channel Source Address (SRC)		QSRC
0200 0008h	Array/frame count (FRMCNT)	Element count (ELECNT)	QCNT
0200 000Ch	QDMA Channel Destination Address (DST)		QDST
0200 0010h	Array/frame index (FRMIDX)	Element index (ELEIDX)	QIDX

(b) QDMA pseudo-registers

Address			QDMA pseudo-register
0200 0020h	QDMA Channel Options		QSOPT
0200 0024h	QDMA Channel Source Address (SRC)		QSSRC
0200 0028h	Array/frame count (FRMCNT)	Element count (ELECNT)	QSCNT
0200 002Ch	QDMA Channel Destination Address (DST)		QSDST
0200 0030h	Array/frame index (FRMIDX)	Element index (ELEIDX)	QSIDX

4.9.1 QDMA Channel Options Register (QOPT, QSOPT)

Figure 4−28. QDMA Channel Options Register (QOPT)

31 29	28 27	26	25 24	23	22 21	20	19 16
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

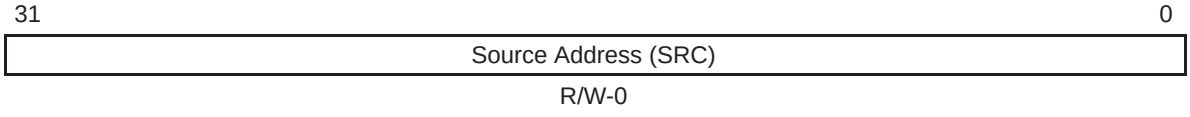
15	14 13	12 1	0
Rsvd	TCCM	Reserved	FS
R/W-0	R/W-0	R/W-0	R/W-0

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

Note: QOPT is read/writable; QSOPT is write-only.

4.9.2 QDMA Channel Source Address Register (QSRC, QSSRC)

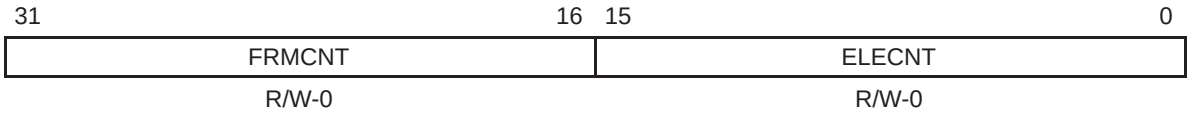
Figure 4−29. QDMA Channel Source Address Register (QSRC)



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

4.9.3 QDMA Channel Transfer Count Register (QCNT, QSCNT)

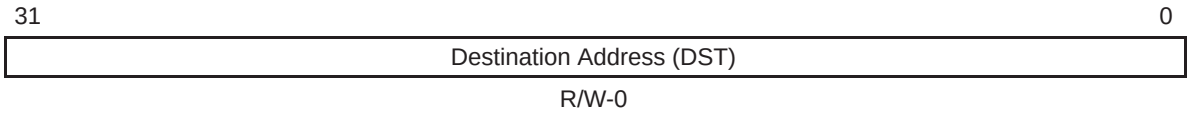
Figure 4−30. QDMA Channel Transfer Count Register (QCNT)



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

4.9.4 QDMA Channel Destination Address Register (QDST, QSDST)

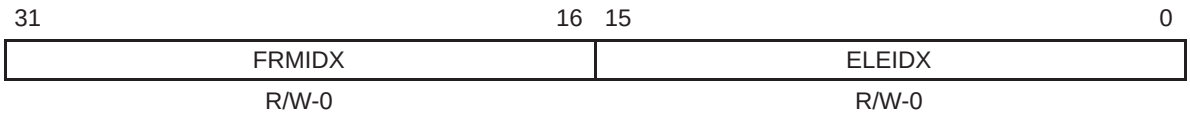
Figure 4−31. QDMA Channel Destination Address Register (QDST)



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

4.9.5 QDMA Channel Index Register (QIDX, QSIDX)

Figure 4−32. QDMA Channel Index Register (QIDX)



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

Appendix A

EDMA Transfers

This appendix describes all of the different types of EDMA transfers.

Topic **Page**

A.1 Element Synchronized 1D-to-1D Transfers

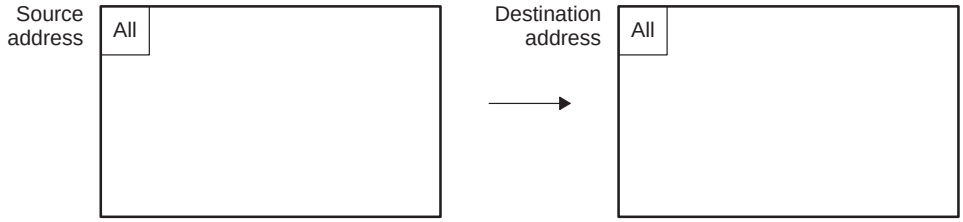
The possible 1D-to-1D transfers (2DS = 2DD = 0), along with the necessary parameters using element synchronization (FS = 0), are listed in Table A−1 and shown in Figure A−1 through Figure A−16. For each, only one element is transferred per synchronization event.

Table A−1. Element Synchronized (FS = 0) 1D-to-1D Transfers

	Channel Options Parameter (OPT)		
Source address	**SUM Bits**	**DUM Bits**	**Figure**
Fixed	00	00	Figure A−1
	00	01	Figure A−2
	00	10	Figure A−3
	00	11	Figure A−4
Incremented	01	00	Figure A−5
	01	01	Figure A−6
	01	10	Figure A−7
	01	11	Figure A−8
Decremented	10	00	Figure A−9
	10	01	Figure A−10
	10	10	Figure A−11
	10	11	Figure A−12
Indexed	11	00	Figure A−13
	11	01	Figure A−14
	11	10	Figure A−15
	11	11	Figure A−16

Figure A−1. Element Synchronized 1D-to-1D Transfer (SUM = 00, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2000 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
0004h	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

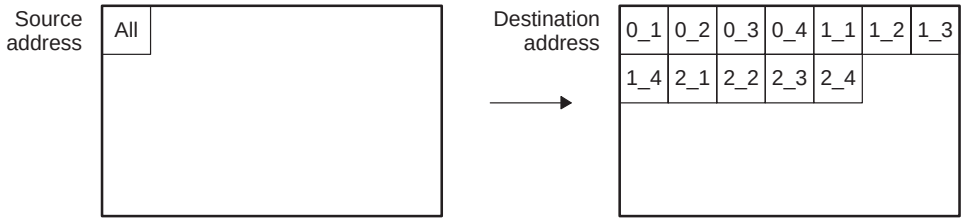
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	00	0	00	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−2. Element Synchronized 1D-to-1D Transfer (SUM = 00, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2020 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
0004h	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

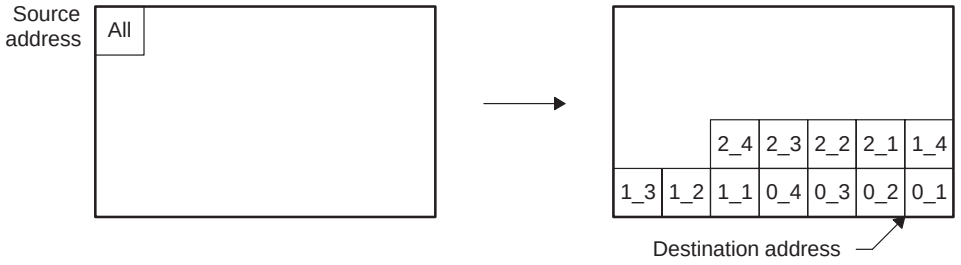
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	00	0	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−3. Element Synchronized 1D-to-1D Transfer (SUM = 00, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2040 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

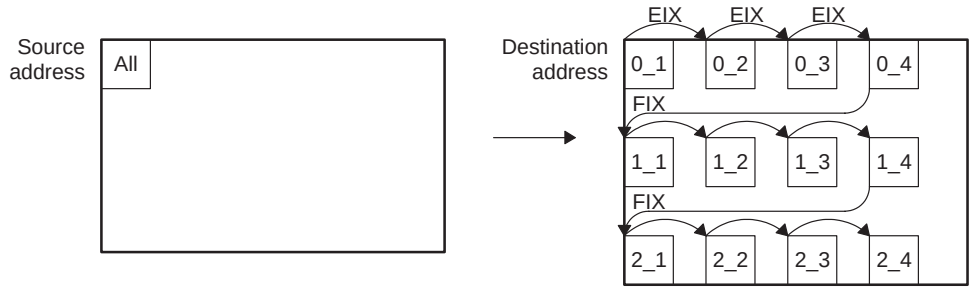
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	00	0	10	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−4. Element Synchronized 1D-to-1D Transfer (SUM = 00, DUM = 11)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2060 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

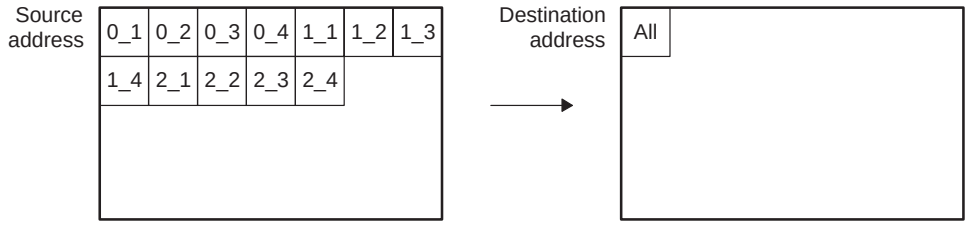
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	00	0	11	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−5. Element Synchronized 1D-to-1D Transfer (SUM = 01, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2100 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

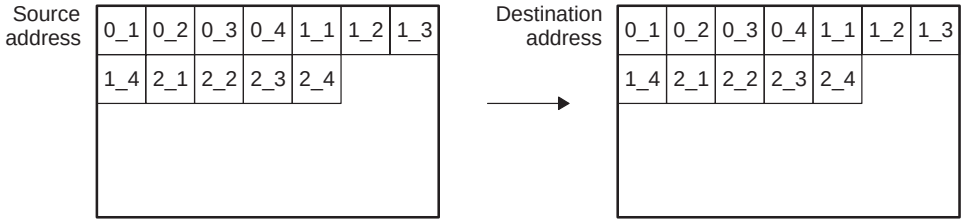
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	01	0	00	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−6. Element Synchronized 1D-to-1D Transfer (SUM = 01, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2120 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
0004h	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

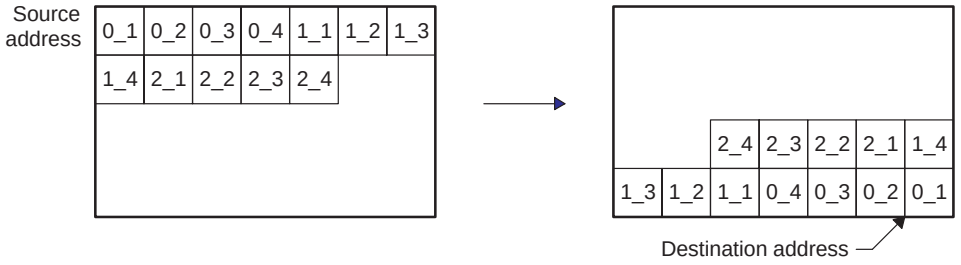
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	01	0	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−7. Element Synchronized 1D-to-1D Transfer (SUM = 01, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2140 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

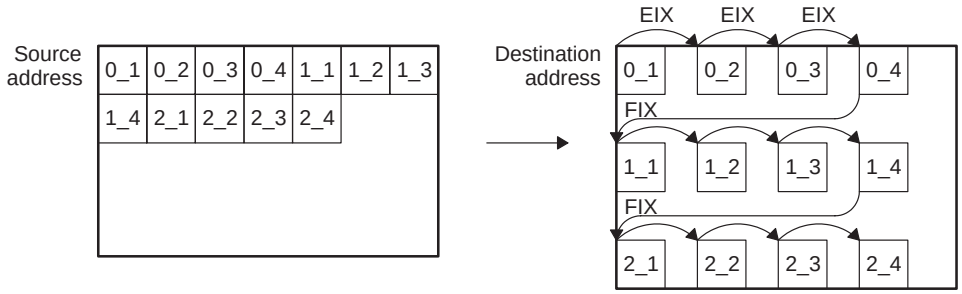
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	01	0	10	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−8. Element Synchronized 1D-to-1D Transfer (SUM = 01, DUM = 11)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2160 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

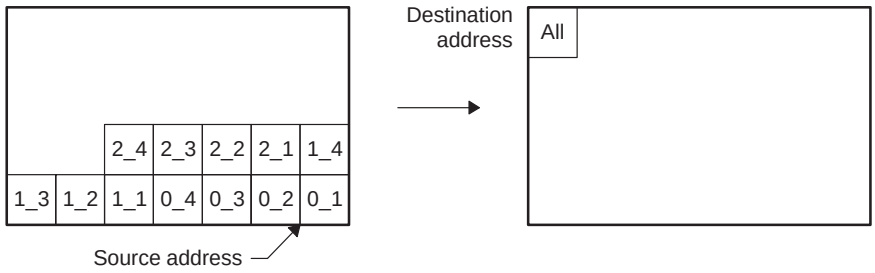
31–29	28–27	26	25–24	23	22–21	20	19–16
001	00	0	01	0	11	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14–13	12–2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−9. Element Synchronized 1D-to-1D Transfer (SUM = 10, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2200 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
0004h	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

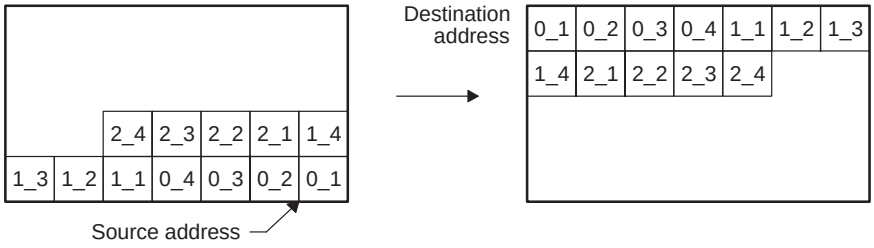
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	10	0	00	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−10. Element Synchronized 1D-to-1D Transfer (SUM = 10, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2220 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

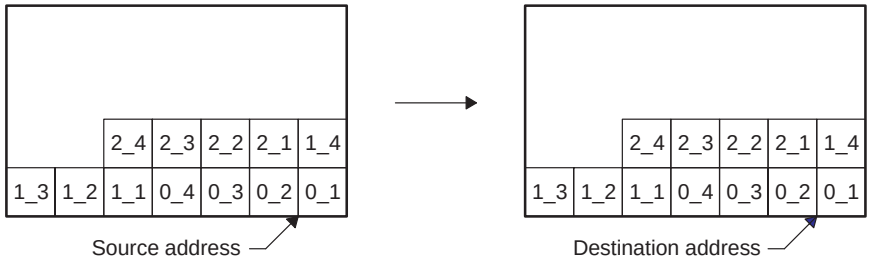
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	10	0	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−11. Element Synchronized 1D-to-1D Transfer (SUM = 10, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2240 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
0004h	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

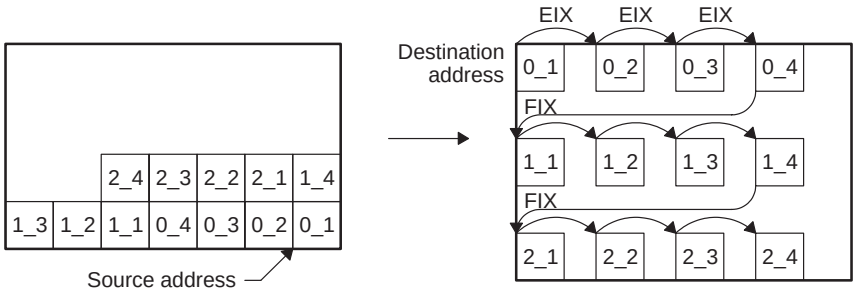
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	10	0	10	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−12. Element Synchronized 1D-to-1D Transfer (SUM = 10, DUM = 11)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2260 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

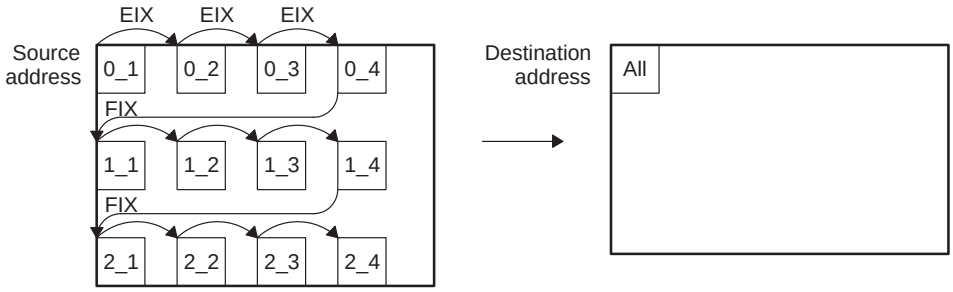
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	10	0	11	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−13. Element Synchronized 1D-to-1D Transfer (SUM = 11, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2300 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

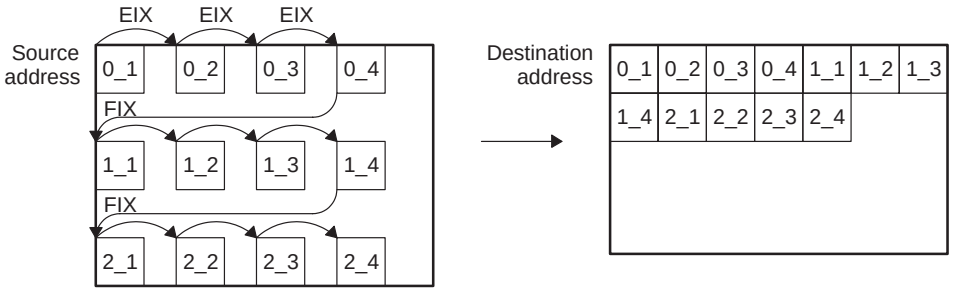
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	11	0	00	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−14. Element Synchronized 1D-to-1D Transfer (SUM = 11, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2320 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

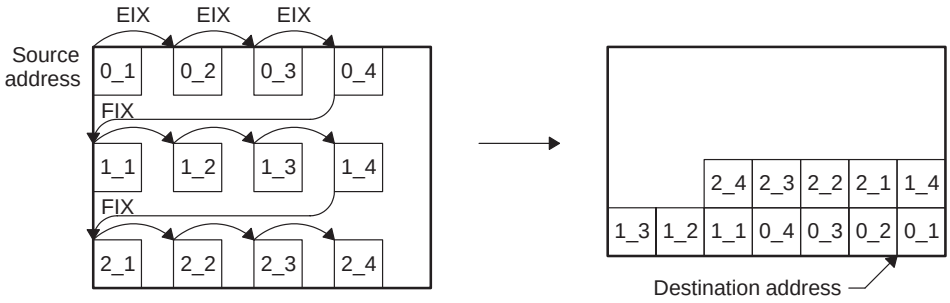
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	11	0	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−15. Element Synchronized 1D-to-1D Transfer (SUM = 11, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2340 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

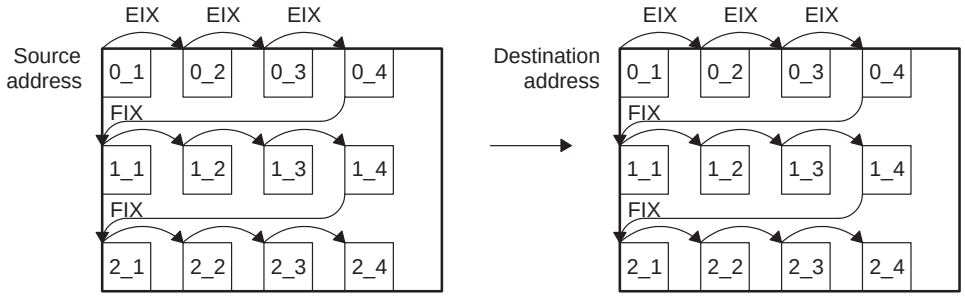
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	11	0	10	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−16. Element Synchronized 1D-to-1D Transfer (SUM = 11, DUM = 11)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2360 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
0004h	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	11	0	11	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

A.2 Frame Synchronized 1D-to-1D Transfers

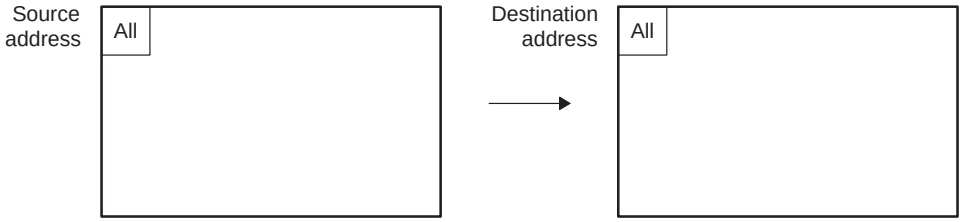
The possible 1D-to-1D transfers (2DS = 2DD = 0), along with the necessary parameters using frame synchronization (FS = 1) are listed in Table A−2 and shown in Figure A−17 through Figure A−32. For each, an entire frame of elements is transferred per synchronization event.

Table A−2. Frame Synchronized (FS = 1) 1D-to-1D Transfers

	Channel Options Parameter (OPT)		
Source address	**SUM Bits**	**DUM Bits**	**Figure**
Fixed	00	00	Figure A−17
	00	01	Figure A−18
	00	10	Figure A−19
	00	11	Figure A−20
Incremented	01	00	Figure A−21
	01	01	Figure A−22
	01	10	Figure A−23
	01	11	Figure A−24
Decremented	10	00	Figure A−25
	10	01	Figure A−26
	10	10	Figure A−27
	10	11	Figure A−28
Indexed	11	00	Figure A−29
	11	01	Figure A−30
	11	10	Figure A−31
	11	11	Figure A−32

Figure A−17. Frame Synchronized 1D-to-1D Transfer (SUM = 00, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2000 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

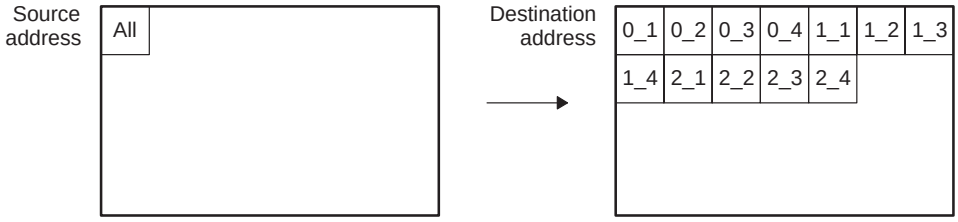
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	00	0	00	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−18. Frame Synchronized 1D-to-1D Transfer (SUM = 00, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2020 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

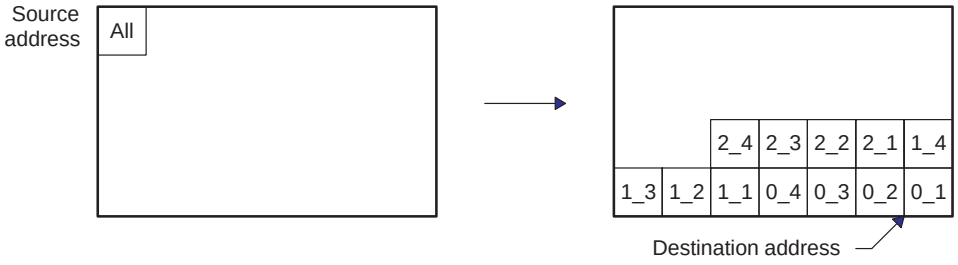
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	00	0	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−19. Frame Synchronized 1D-to-1D Transfer (SUM = 00, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2040 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

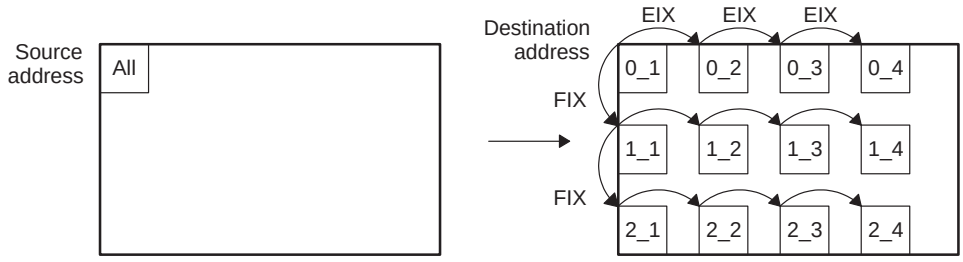
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	00	0	10	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−20. Frame Synchronized 1D-to-1D Transfer (SUM = 00, DUM = 11)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2060 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

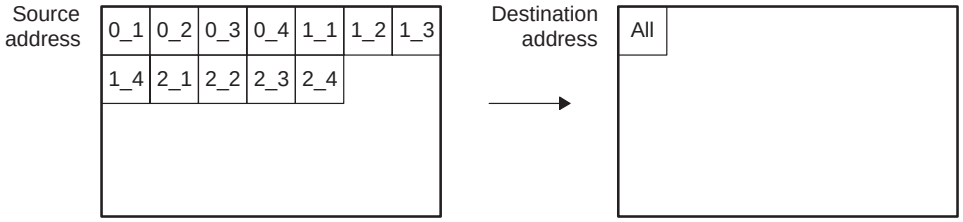
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	00	0	11	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−21. Frame Synchronized 1D-to-1D Transfer (SUM = 01, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2100 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

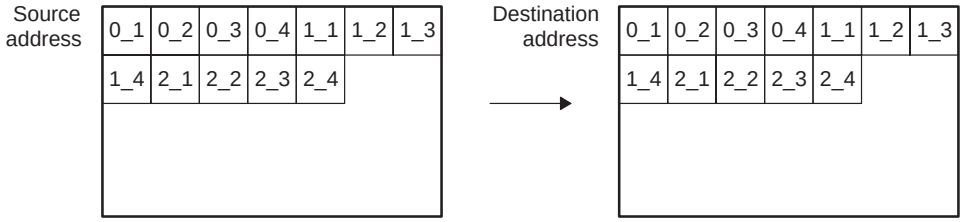
31–29	28–27	26	25–24	23	22–21	20	19–16
001	00	0	01	0	00	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14–13	12–2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−22. Frame Synchronized 1D-to-1D Transfer (SUM = 01, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2120 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

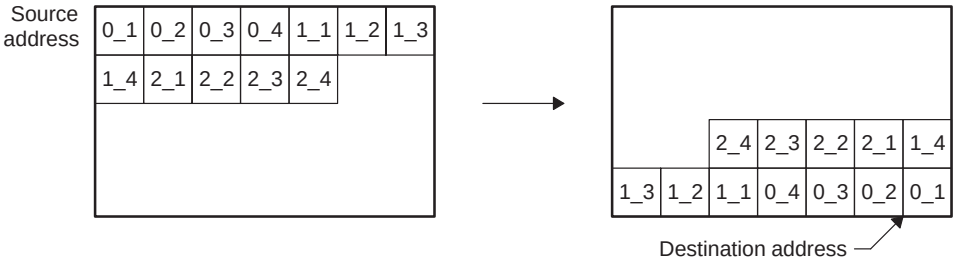
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	01	0	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−23. Frame Synchronized 1D-to-1D Transfer (SUM = 01, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2140 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

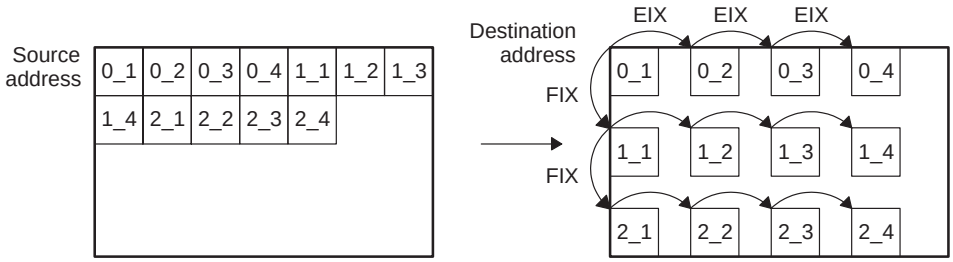
31–29	28–27	26	25–24	23	22–21	20	19–16
001	00	0	01	0	10	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14–13	12–2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−24. Frame Synchronized 1D-to-1D Transfer (SUM = 01, DUM = 11)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2160 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

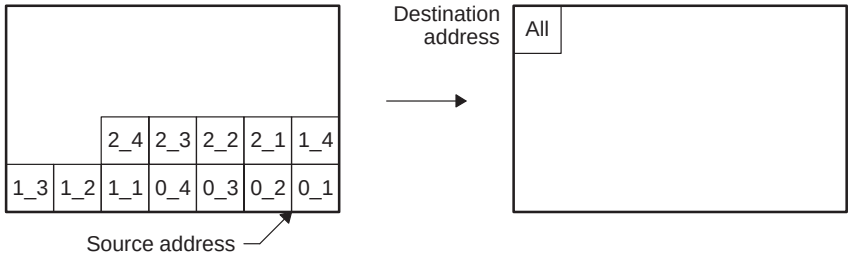
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	01	0	11	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−25. Frame Synchronized 1D-to-1D Transfer (SUM = 10, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2200 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

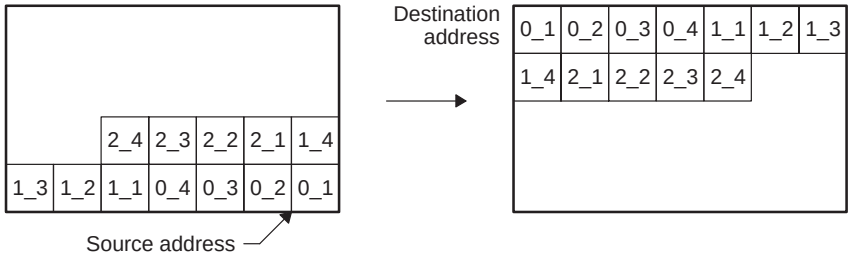
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	10	0	00	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−26. Frame Synchronized 1D-to-1D Transfer (SUM = 10, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2220 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

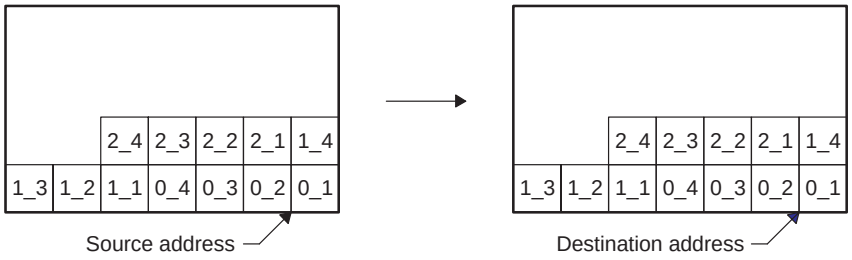
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	10	0	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−27. Frame Synchronized 1D-to-1D Transfer (SUM = 10, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2240 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

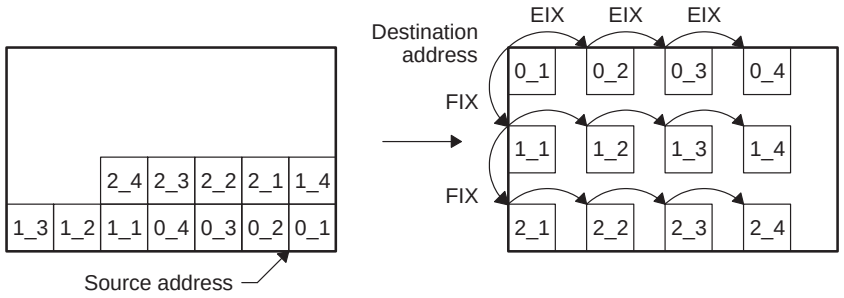
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	10	0	10	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−28. Frame Synchronized 1D-to-1D Transfer (SUM = 10, DUM = 11)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2260 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

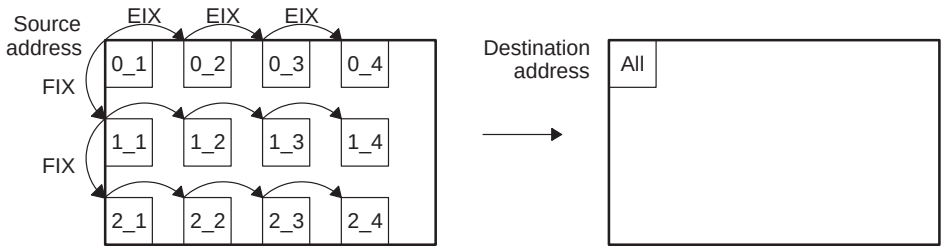
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	10	0	11	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−29. Frame Synchronized 1D-to-1D Transfer (SUM = 11, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2300 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

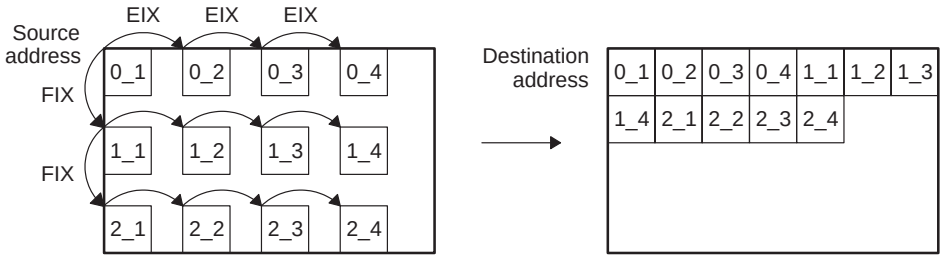
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	11	0	00	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−30. Frame Synchronized 1D-to-1D Transfer (SUM = 11, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2320 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

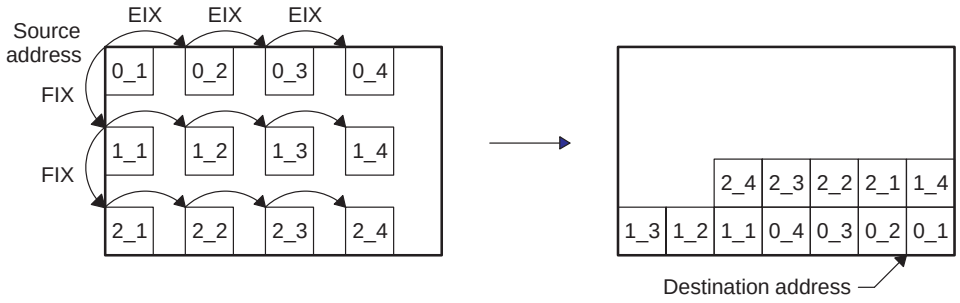
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	11	0	01	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−31. Frame Synchronized 1D-to-1D Transfer (SUM = 11, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2340 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

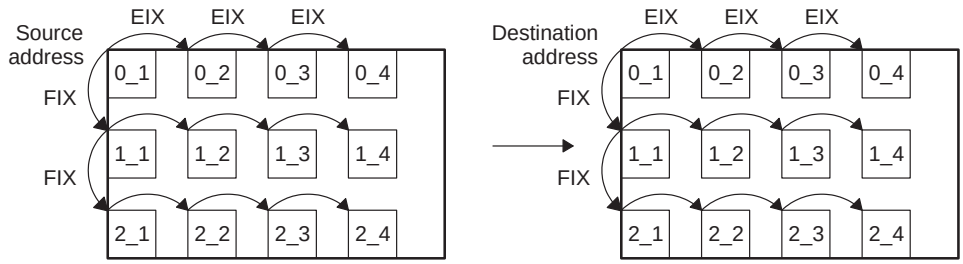
31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	11	0	10	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

Figure A−32. Frame Synchronized 1D-to-1D Transfer (SUM = 11, DUM = 11)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
2360 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
FIX (frame index)	EIX (element index)	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
001	00	0	11	0	11	0	0000
PRI	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM†	Reserved	LINK	FS

† TCCM is reserved on C621x/C671x DSP.

A.3 Array Synchronized 2D-to-2D Transfers

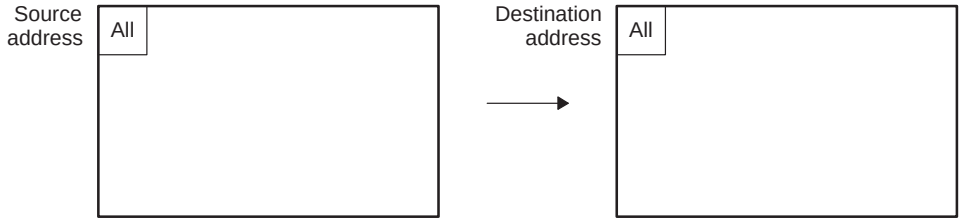
The possible 2D-to-2D transfers (2DS = 2DD = 1), along with the necessary parameters using array synchronization (FS = 0), are listed in Table A−3 and shown in Figure A−33 through Figure A−41. For each, a single array of elements is transferred per synchronization event.

Table A−3. Array Synchronized (FS = 0) 2D-to-2D Transfers

	Channel Options Parameter (OPT)		
Source address	**SUM Bits**	**DUM Bits**	**Figure**
Fixed	00	00	Figure A−33
	00	01	Figure A−34
	00	10	Figure A−35
Incremented	01	00	Figure A−36
	01	01	Figure A−37
	01	10	Figure A−38
Decremented	10	00	Figure A−39
	10	01	Figure A−40
	10	10	Figure A−41

Figure A−33. Array Synchronized 2D-to-2D Transfer (SUM = 00, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4480 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

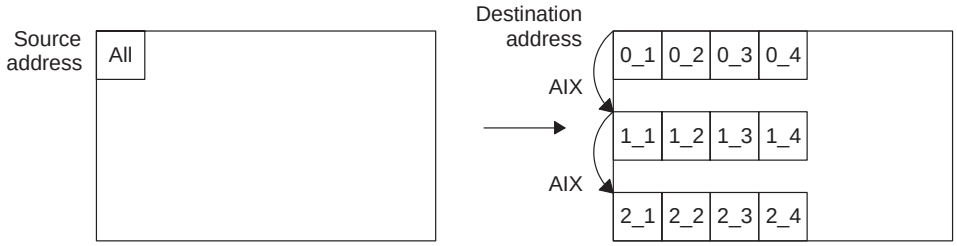
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−34. Array Synchronized 2D-to-2D Transfer (SUM = 00, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
44A0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

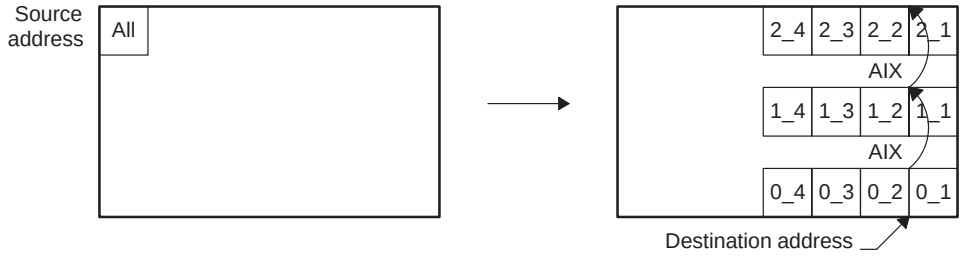
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−35. Array Synchronized 2D-to-2D Transfer (SUM = 00, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
44C0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

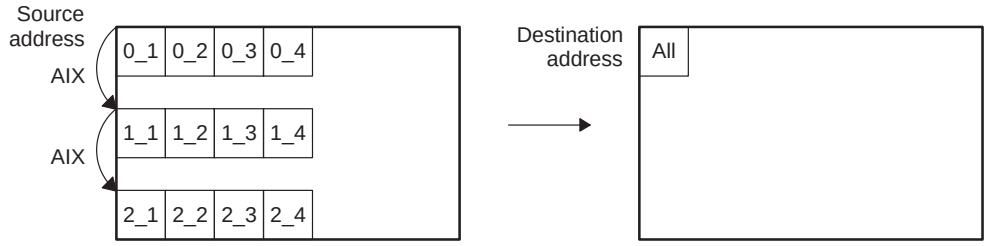
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−36. Array Synchronized 2D-to-2D Transfer (SUM = 01, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4580 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

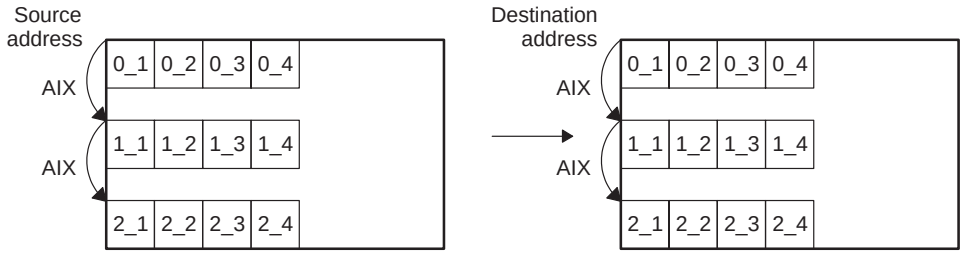
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−37. Array Synchronized 2D-to-2D Transfer (SUM = 01, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
45A0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

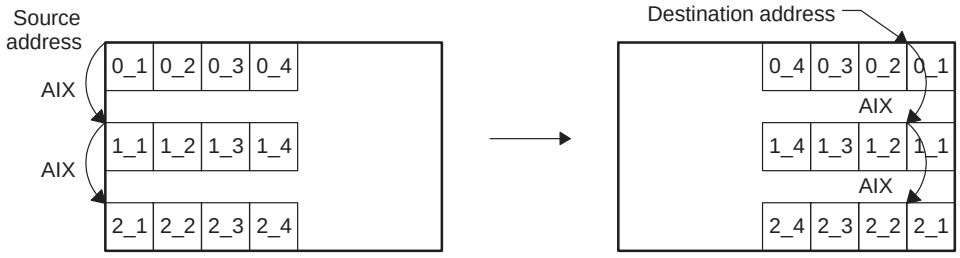
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−38. Array Synchronized 2D-to-2D Transfer (SUM = 01, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
45C0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

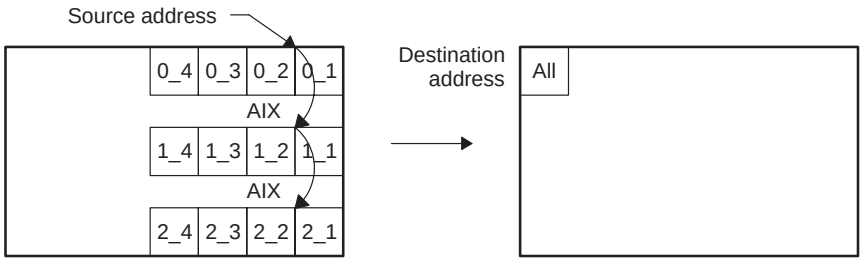
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−39. Array Synchronized 2D-to-2D Transfer (SUM = 10, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4680 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

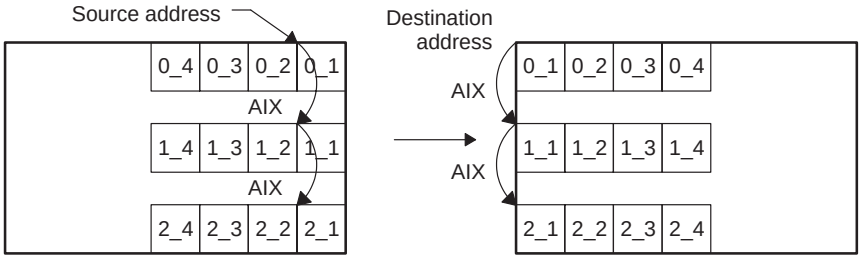
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	10	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−40. Array Synchronized 2D-to-2D Transfer (SUM = 10, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
46A0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

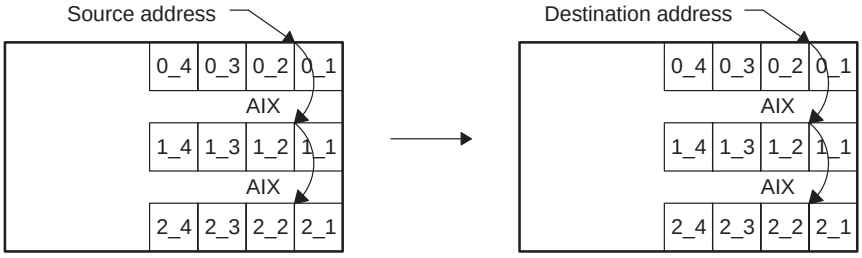
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	10	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−41. Array Synchronized 2D-to-2D Transfer (SUM = 10, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
46C0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	10	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

A.4 Block Synchronized 2D-to-2D Transfers

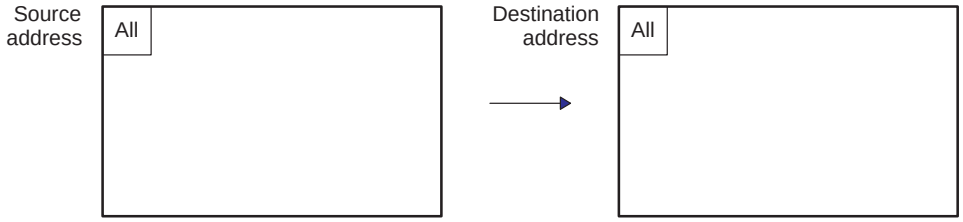
The possible 2D-to-2D transfers (2DS = 2DD = 1), along with the necessary parameters using block synchronization (FS = 1), are listed in Table A−4 and shown in Figure A−42 through Figure A−50. For each, an entire block of arrays is transferred per synchronization event.

Table A−4. Block Synchronized (FS = 1) 2D-to-2D Transfers

	Channel Options Parameter (OPT)		
Source address	**SUM Bits**	**DUM Bits**	**Figure**
Fixed	00	00	Figure A−42
	00	01	Figure A−43
	00	10	Figure A−44
Incremented	01	00	Figure A−45
	01	01	Figure A−46
	01	10	Figure A−47
Decremented	10	00	Figure A−48
	10	01	Figure A−49
	10	10	Figure A−50

Figure A−42. Block Synchronized 2D-to-2D Transfer (SUM = 00, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4480 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

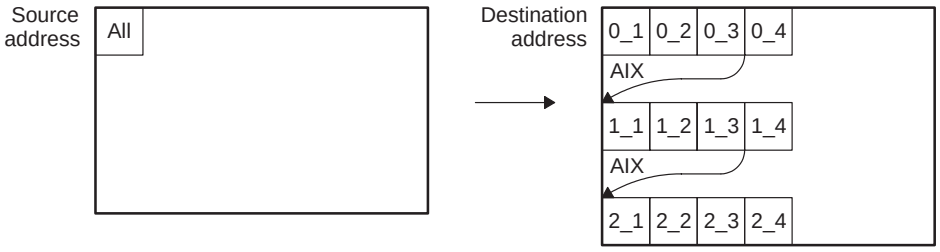
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−43. Block Synchronized 2D-to-2D Transfer (SUM = 00, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
44A0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

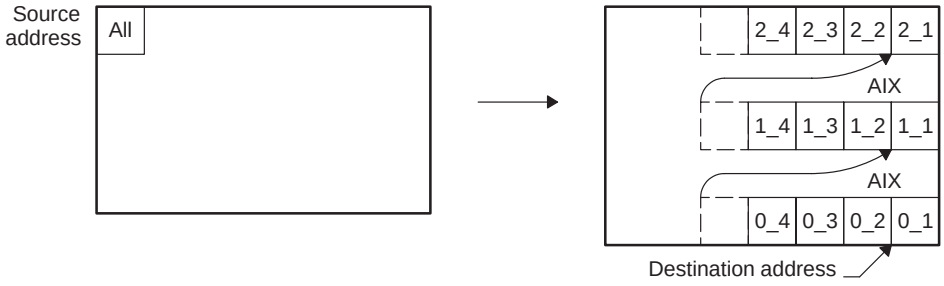
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−44. Block Synchronized 2D-to-2D Transfer (SUM = 00, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
44C0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

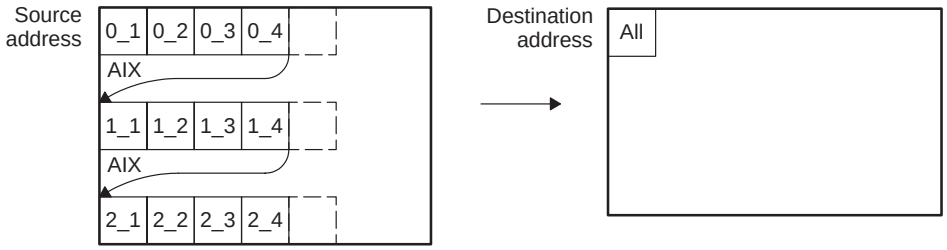
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−45. Block Synchronized 2D-to-2D Transfer (SUM = 01, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4580 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

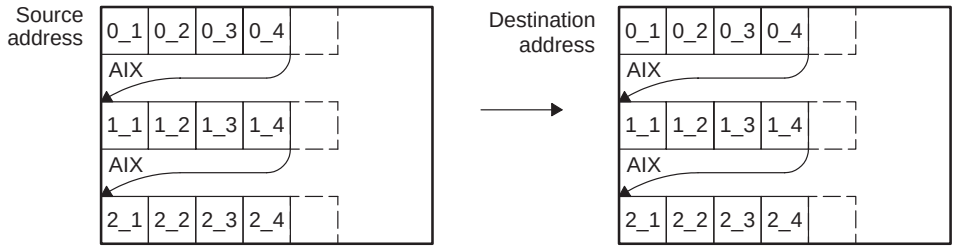
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−46. Block Synchronized 2D-to-2D Transfer (SUM = 01, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
45A0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

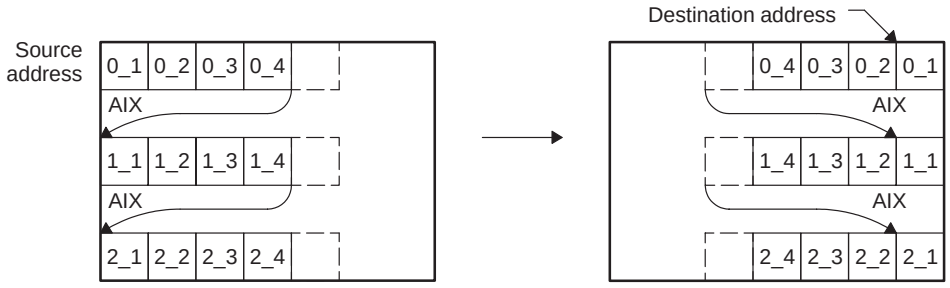
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−47. Block Synchronized 2D-to-2D Transfer (SUM = 01, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
45C0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

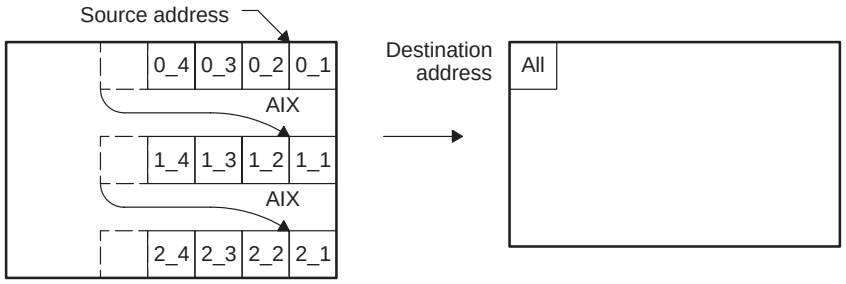
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−48. Block Synchronized 2D-to-2D Transfer (SUM = 10, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4680 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

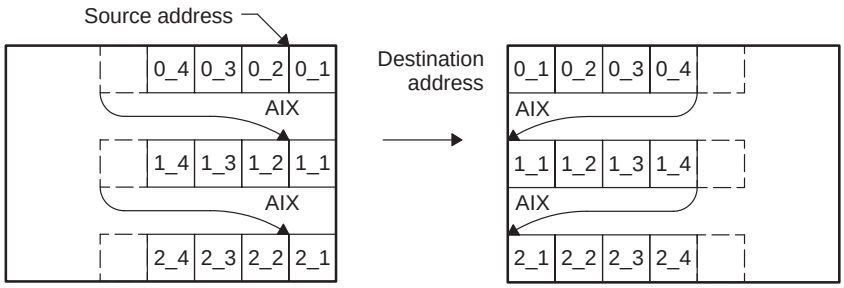
31–29	28–27	26	25–24	23	22–21	20	19–16
010	00	1	10	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14–13	12–2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−49. Block Synchronized 2D-to-2D Transfer (SUM = 10, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
46A0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

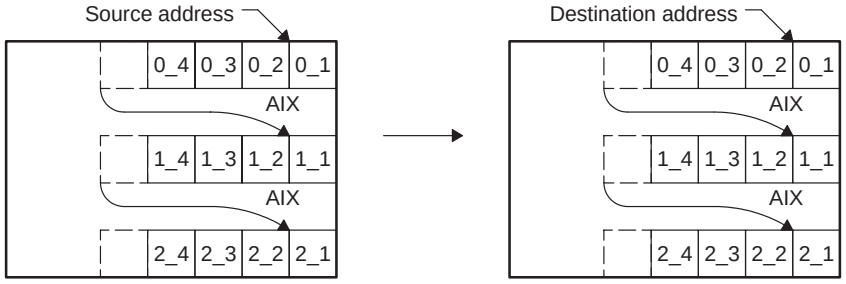
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	10	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−50. Block Synchronized 2D-to-2D Transfer (SUM = 10, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
46C0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	10	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

A.5 Array Synchronized 1D-to-2D Transfers

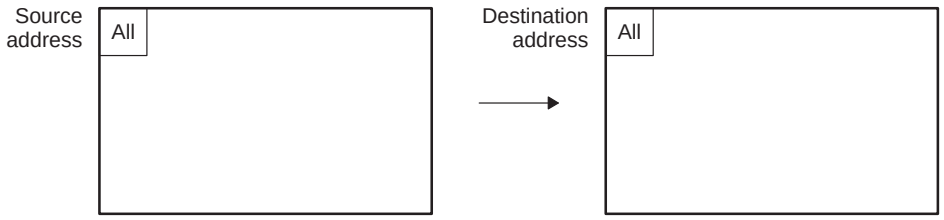
The possible 1D-to-2D transfers (2DS = 0, 2DD = 1), along with the necessary parameters using array synchronization (FS = 0), are listed in Table A−5 and shown in Figure A−51 through Figure A−59. For each, a single array of elements is transferred per synchronization event.

Table A−5. Array Synchronized (FS = 0) 1D-to-2D Transfers

	Channel Options Parameter (OPT)		
Source address	**SUM Bits**	**DUM Bits**	**Figure**
Fixed	00	00	Figure A−51
	00	01	Figure A−52
	00	10	Figure A−53
Incremented	01	00	Figure A−54
	01	01	Figure A−55
	01	10	Figure A−56
Decremented	10	00	Figure A−57
	10	01	Figure A−58
	10	10	Figure A−59

Figure A−51. Array Synchronized 1D-to-2D Transfer (SUM = 00, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4080 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

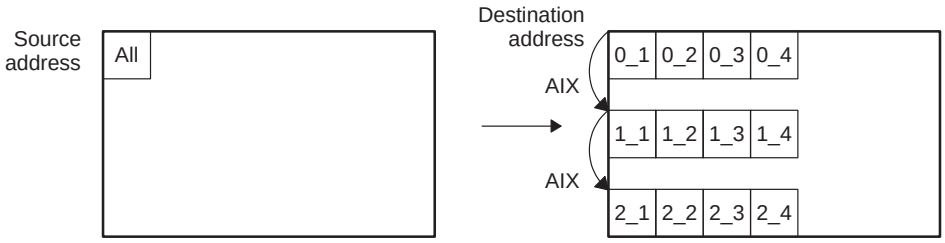
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	00	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−52. Array Synchronized 1D-to-2D Transfer (SUM = 00, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
40A0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

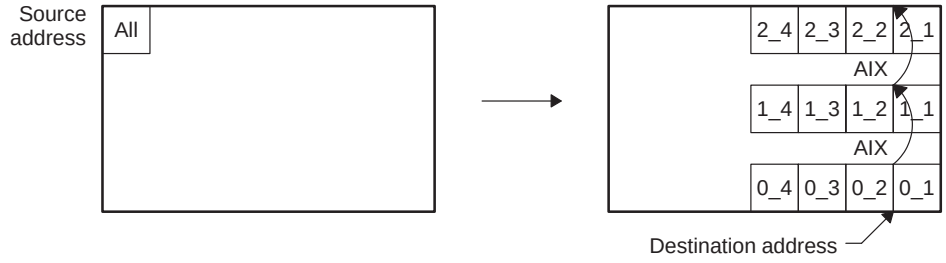
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	00	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−53. Array Synchronized 1D-to-2D Transfer (SUM = 00, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
40C0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

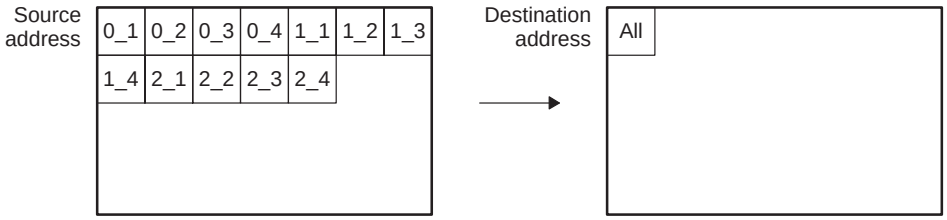
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	00	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−54. Array Synchronized 1D-to-2D Transfer (SUM = 01, DUM = 00)

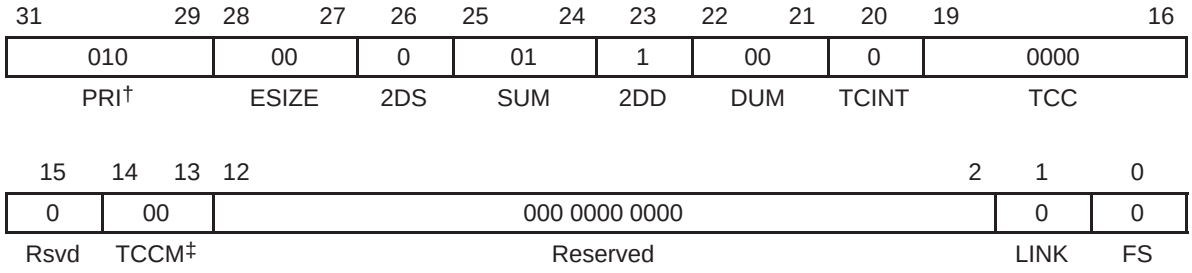
(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4180 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

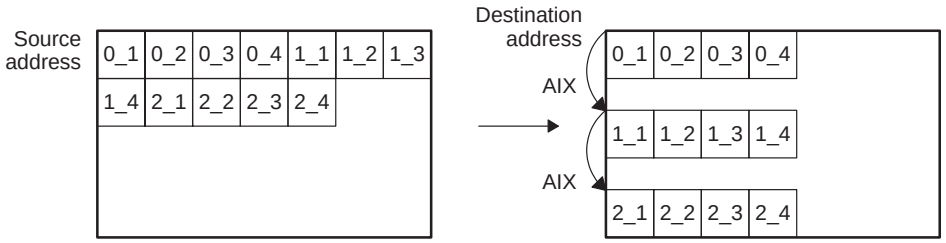
(c) EDMA Channel Options Parameter (OPT) Content



† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−55. Array Synchronized 1D-to-2D Transfer (SUM = 01, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
41A0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

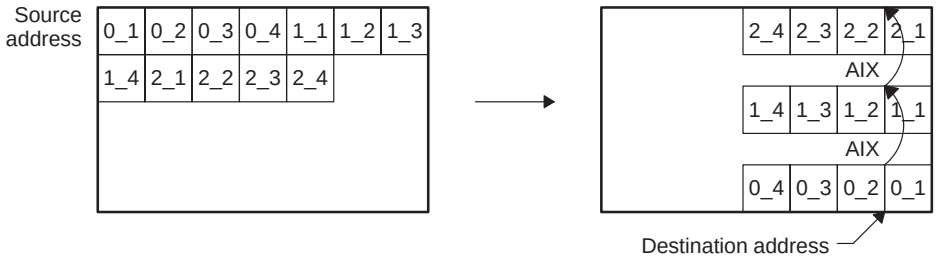
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	01	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−56. Array Synchronized 1D-to-2D Transfer (SUM = 01, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
41C0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

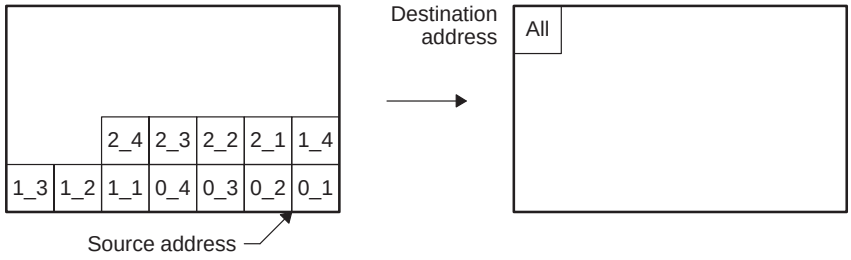
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	01	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−57. Array Synchronized 1D-to-2D Transfer (SUM = 10, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4280 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

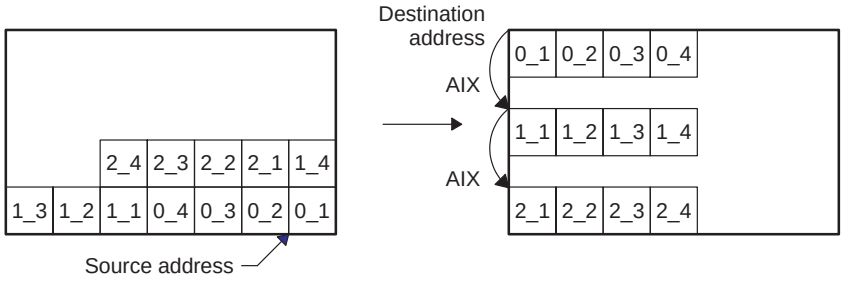
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	10	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−58. Array Synchronized 1D-to-2D Transfer (SUM = 10, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
42A0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

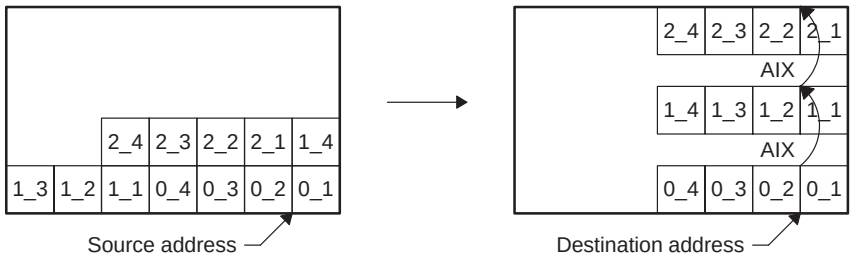
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	10	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−59. Array Synchronized 1D-to-2D Transfer (SUM = 10, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
42C0 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	10	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

A.6 Block Synchronized 1D-to-2D Transfers

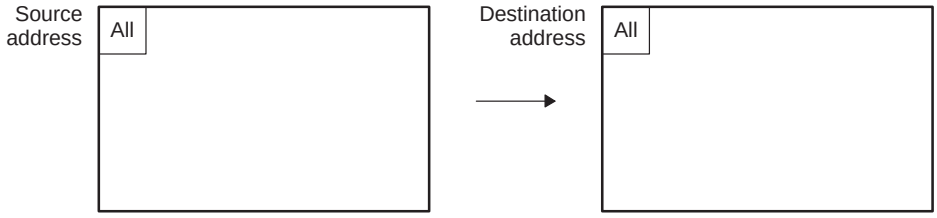
The possible 1D-to-2D transfers (2DS = 0, 2DD = 1), along with the necessary parameters using block synchronization (FS = 1), are listed in Table A−6 and shown in Figure A−60 through Figure A−68. For each, an entire block of arrays is transferred per synchronization event.

Table A−6. Block Synchronized (FS = 1) 1D-to-2D Transfers

	Channel Options Parameter (OPT)		
Source address	**SUM Bits**	**DUM Bits**	**Figure**
Fixed	00	00	Figure A−60
	00	01	Figure A−61
	00	10	Figure A−62
Incremented	01	00	Figure A−63
	01	01	Figure A−64
	01	10	Figure A−65
Decremented	10	00	Figure A−66
	10	01	Figure A−67
	10	10	Figure A−68

Figure A−60. Block Synchronized 1D-to-2D Transfer (SUM = 00, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4080 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

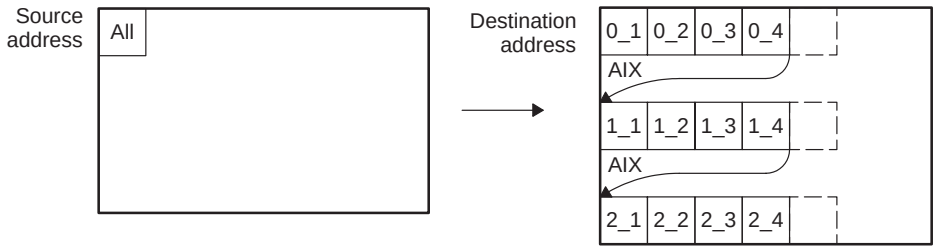
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	00	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−61. Block Synchronized 1D-to-2D Transfer (SUM = 00, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
40A0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

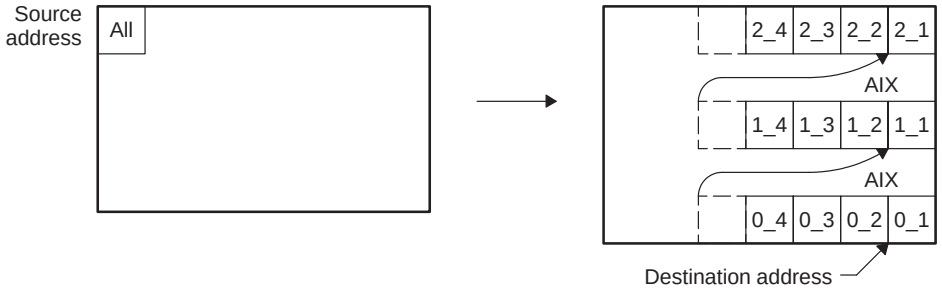
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	00	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−62. Block Synchronized 1D-to-2D Transfer (SUM = 00, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
40C0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

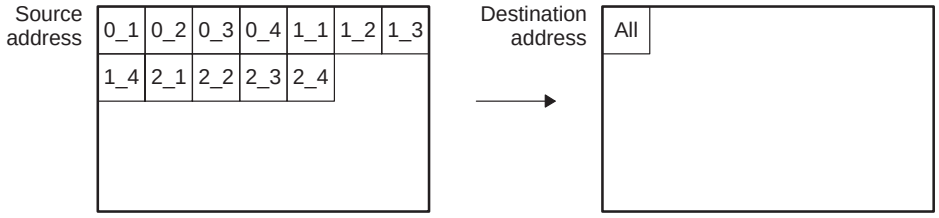
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	00	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−63. Block Synchronized 1D-to-2D Transfer (SUM = 01, DUM = 00)

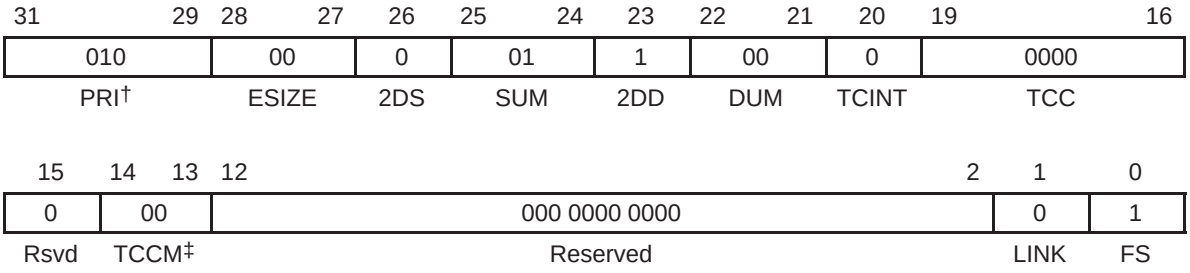
(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4180 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

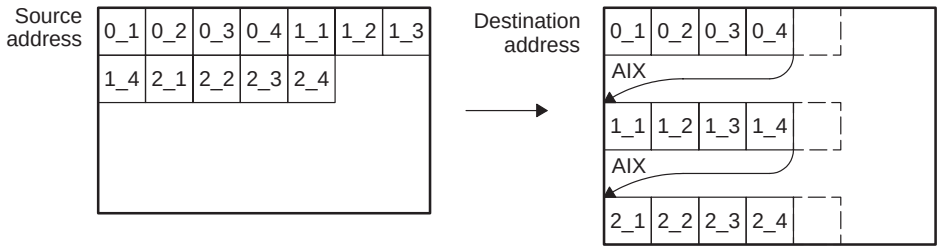
(c) EDMA Channel Options Parameter (OPT) Content



† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−64. Block Synchronized 1D-to-2D Transfer (SUM = 01, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
41A0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

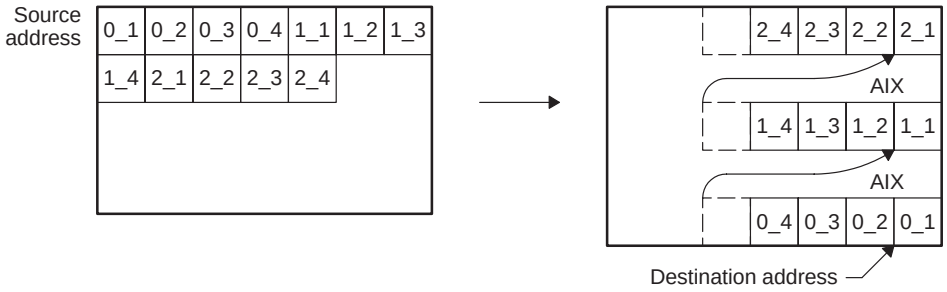
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	01	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−65. Block Synchronized 1D-to-2D Transfer (SUM = 01, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
41C0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

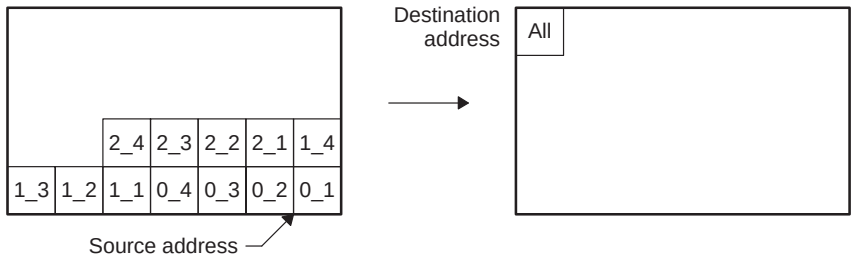
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	01	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−66. Block Synchronized 1D-to-2D Transfer (SUM = 10, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4280 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

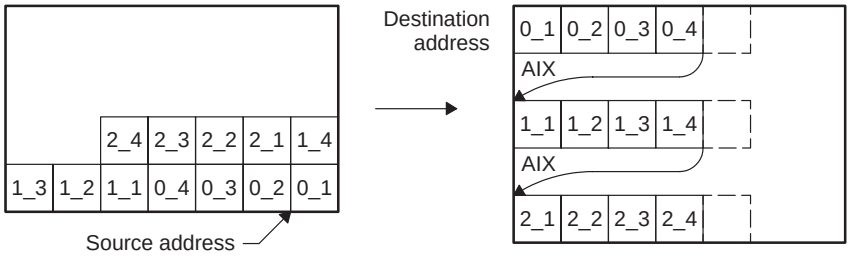
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	10	1	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−67. Block Synchronized 1D-to-2D Transfer (SUM = 10, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
42A0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

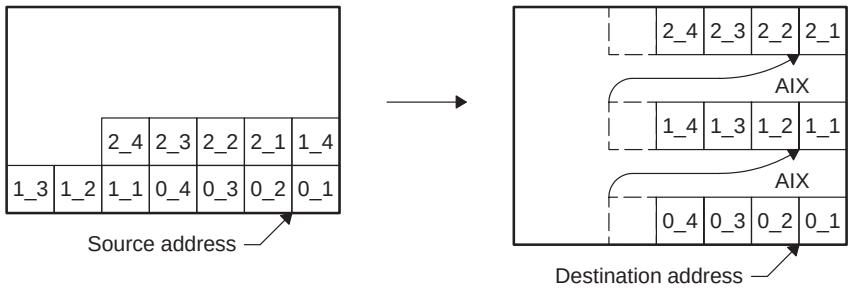
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	10	1	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−68. Block Synchronized 1D-to-2D Transfer (SUM = 10, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
42C0 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	0	10	1	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15 14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

A.7 Array Synchronized 2D-to-1D Transfers

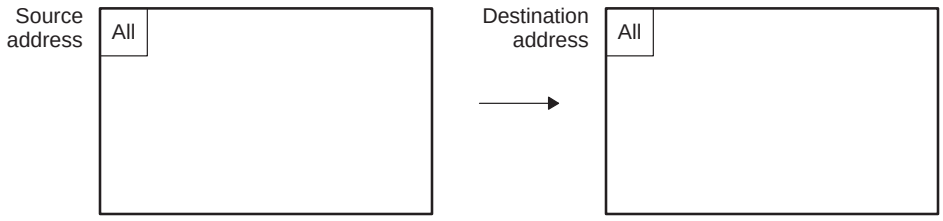
The possible 2D-to-1D transfers (2DS = 1, 2DD = 0), along with the necessary parameters using array synchronization (FS = 0), are listed in Table A−7 and shown in Figure A−69 through Figure A−77. For each, a single array of elements is transferred per synchronization event.

Table A−7. Array Synchronized (FS = 0) 2D-to-1D Transfers

	Channel Options Parameter (OPT)		
Source address	**SUM Bits**	**DUM Bits**	**Figure**
Fixed	00	00	Figure A−69
	00	01	Figure A−70
	00	10	Figure A−71
Incremented	01	00	Figure A−72
	01	01	Figure A−73
	01	10	Figure A−74
Decremented	10	00	Figure A−75
	10	01	Figure A−76
	10	10	Figure A−77

Figure A−69. Array Synchronized 2D-to-1D Transfer (SUM = 00, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4400 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

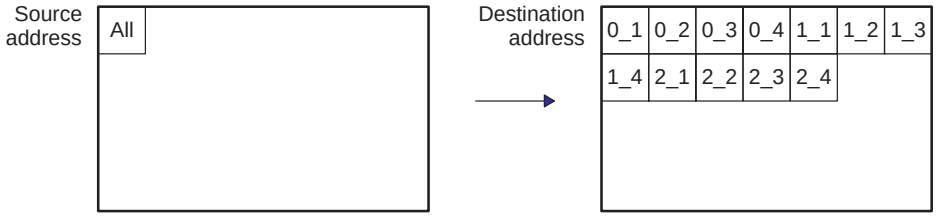
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	0	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−70. Array Synchronized 2D-to-1D Transfer (SUM = 00, DUM = 01)

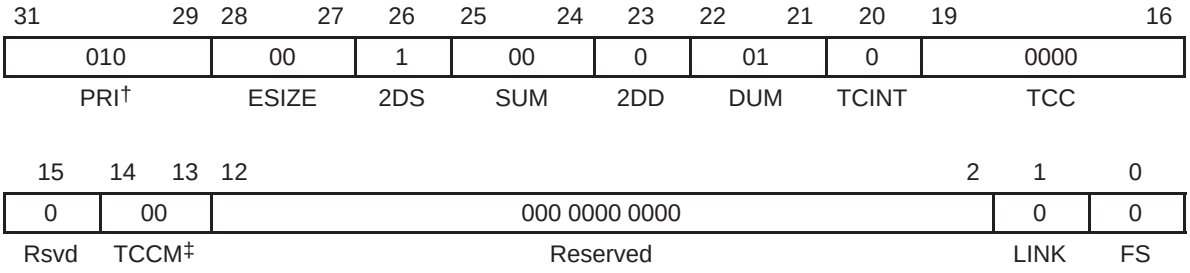
(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4420 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

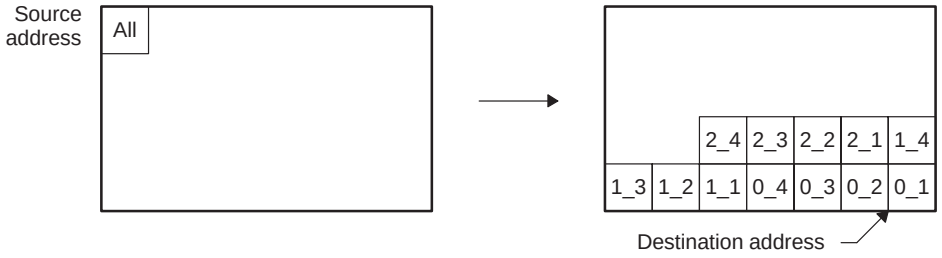
(c) EDMA Channel Options Parameter (OPT) Content



† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−71. Array Synchronized 2D-to-1D Transfer (SUM = 00, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4440 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

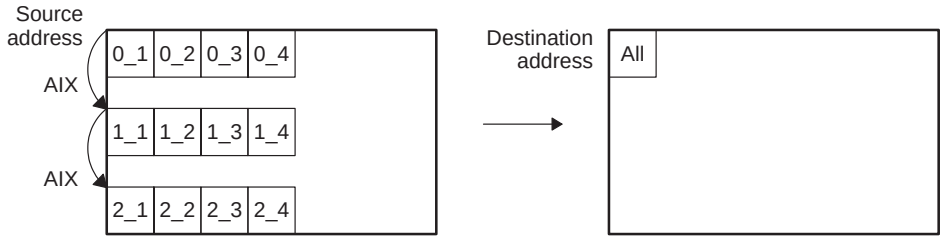
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	0	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−72. Array Synchronized 2D-to-1D Transfer (SUM = 01, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4500 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

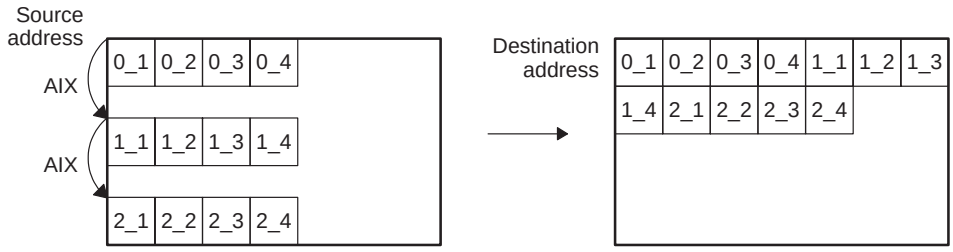
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	0	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−73. Array Synchronized 2D-to-1D Transfer (SUM = 01, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4520 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

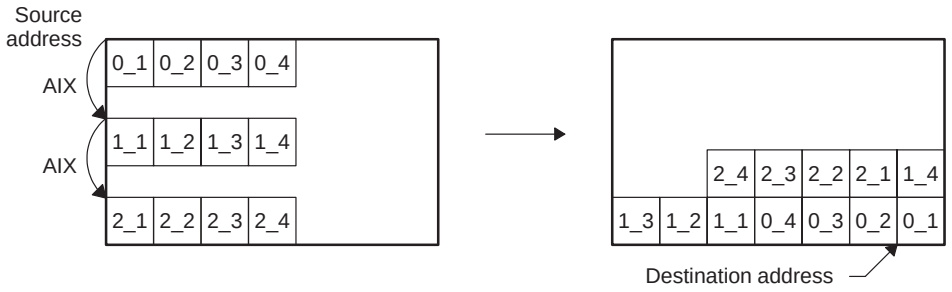
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	0	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−74. Array Synchronized 2D-to-1D Transfer (SUM = 01, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4540 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

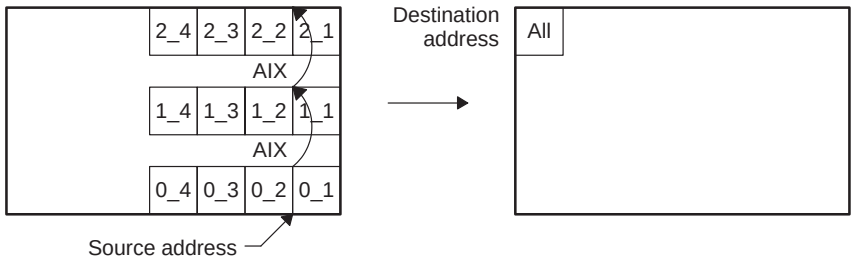
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	0	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−75. Array Synchronized 2D-to-1D Transfer (SUM = 10, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4600 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

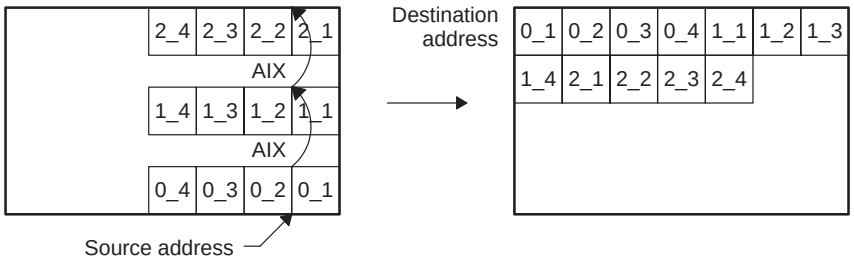
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	10	0	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−76. Array Synchronized 2D-to-1D Transfer (SUM = 10, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4620 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

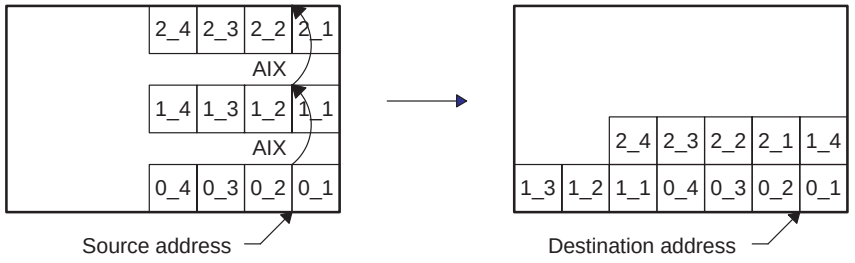
31–29	28–27	26	25–24	23	22–21	20	19–16
010	00	1	10	0	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14–13	12–2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−77. Array Synchronized 2D-to-1D Transfer (SUM = 10, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4640 0000h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	10	0	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	0
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

A.8 Block Synchronized 2D-to-1D Transfers

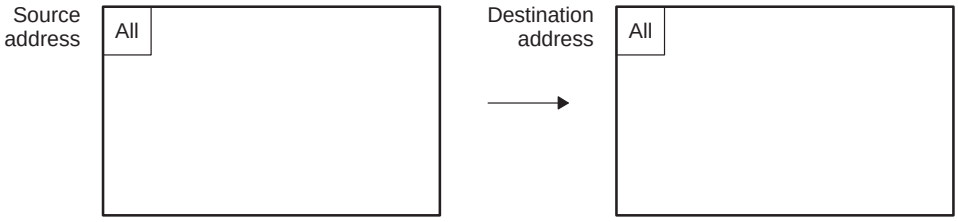
The possible 2D-to-1D transfers (2DS = 1, 2DD = 0), along with the necessary parameters using block synchronization (FS = 1), are listed in Table A−8 and shown in Figure A−78 through Figure A−86. For each, an entire block of arrays is transferred per synchronization event.

Table A−8. Block Synchronized (FS = 1) 2D-to-1D Transfers

	Channel Options Parameter (OPT)		
Source address	**SUM Bits**	**DUM Bits**	**Figure**
Fixed	00	00	Figure A−78
	00	01	Figure A−79
	00	10	Figure A−80
Incremented	01	00	Figure A−81
	01	01	Figure A−82
	01	10	Figure A−83
Decremented	10	00	Figure A−84
	10	01	Figure A−85
	10	10	Figure A−86

Figure A−78. Block Synchronized 2D-to-1D Transfer (SUM = 00, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4400 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

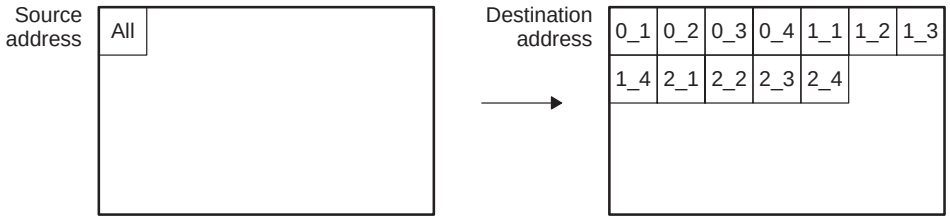
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	0	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−79. Block Synchronized 2D-to-1D Transfer (SUM = 00, DUM = 01)

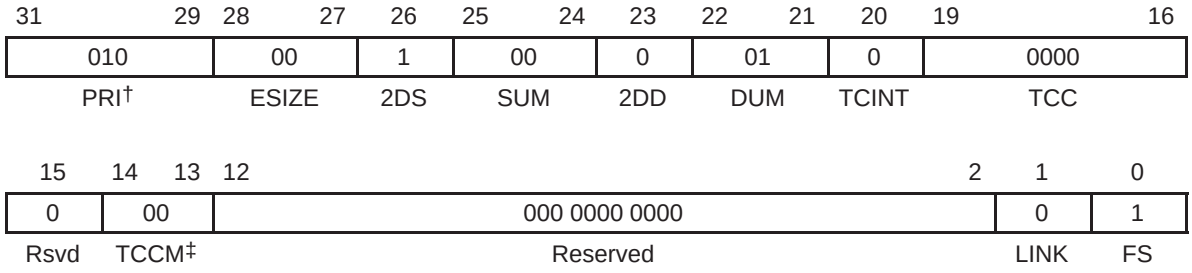
(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4420 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don't care	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

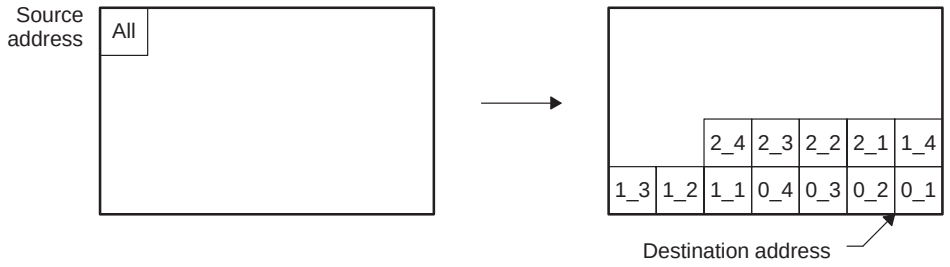
(c) EDMA Channel Options Parameter (OPT) Content



† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−80. Block Synchronized 2D-to-1D Transfer (SUM = 00, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4440 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
Don’t care	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

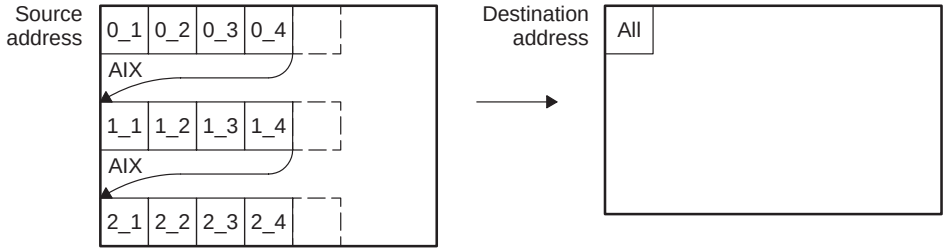
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	00	0	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−81. Block Synchronized 2D-to-1D Transfer (SUM = 01, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4500 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

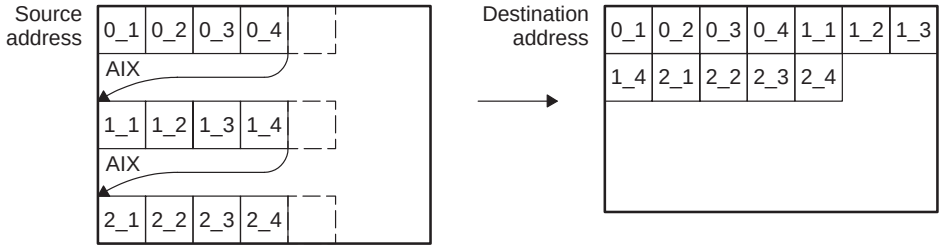
31–29	28–27	26	25–24	23	22–21	20	19–16
010	00	1	01	0	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14–13	12–2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−82. Block Synchronized 2D-to-1D Transfer (SUM = 01, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4520 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

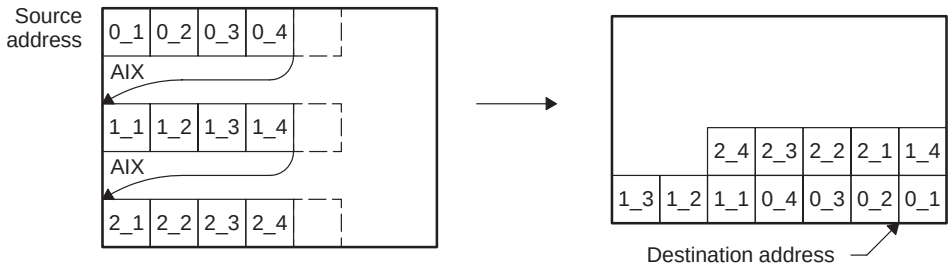
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	0	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−83. Block Synchronized 2D-to-1D Transfer (SUM = 01, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4540 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)	Don’t care	EDMA Channel Index (IDX)
Don’t care	Don’t care	EDMA Channel Count Reload/Link Address (RLD)

(c) EDMA Channel Options Parameter (OPT) Content

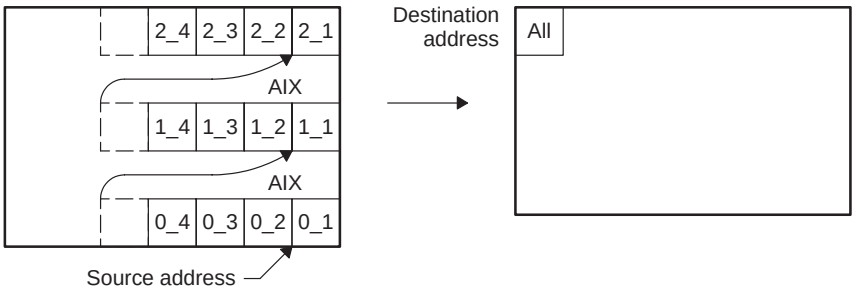
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	01	0	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−84. Block Synchronized 2D-to-1D Transfer (SUM = 10, DUM = 00)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4600 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

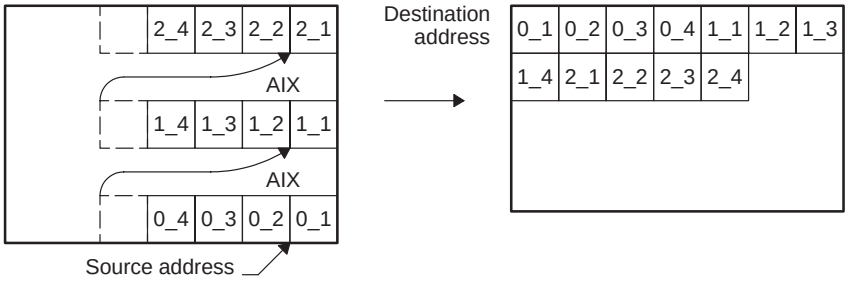
31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	10	0	00	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−85. Block Synchronized 2D-to-1D Transfer (SUM = 10, DUM = 01)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4620 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

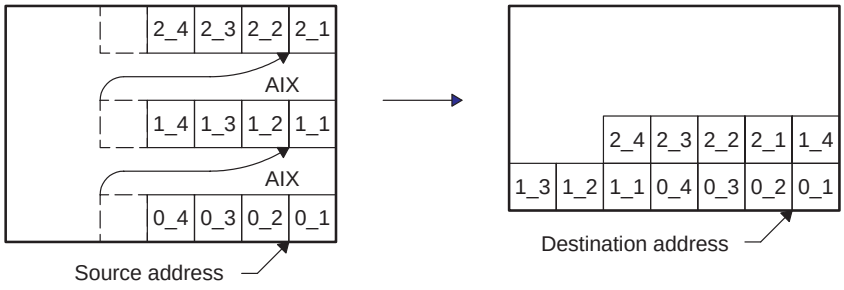
31–29	28–27	26	25–24	23	22–21	20	19–16
010	00	1	10	0	01	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14–13	12–2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Figure A−86. Block Synchronized 2D-to-1D Transfer (SUM = 10, DUM = 10)

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter
4640 0001h		EDMA Channel Options Parameter (OPT)
Source address		EDMA Channel Source Address (SRC)
0002h	0004h	EDMA Channel Transfer Count (CNT)
Destination address		EDMA Channel Destination Address (DST)
AIX (array index)†	Don't care	EDMA Channel Index (IDX)
Don't care	Don't care	EDMA Channel Count Reload/Link Address (RLD)

† AIX is a negative value in this example.

(c) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
010	00	1	10	0	10	0	0000
PRI†	ESIZE	2DS	SUM	2DD	DUM	TCINT	TCC

15	14 13	12 2	1	0
0	00	000 0000 0000	0	1
Rsvd	TCCM‡	Reserved	LINK	FS

† PRI is set to 010b for C621x/C671x DSP or 011b for C64x DSP to select low priority background transfer.
‡ TCCM is reserved on C621x/C671x DSP.

Appendix B

Revision History

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

Table B−1. Document Revision History

Page	Additions/Modifications/Deletions
3-15	Changed read/write value for Figure 3−6
4-16	Changed read/write value for Figure 4−5
4-17	Changed read/write value for Figure 4−6
4-26	Changed read/write value for Figure 4−15
4-27	Changed read/write value for Figure 4−16
4-28	Changed read/write value for Figure 4−17
4-29	Changed read/write value for Figure 4−18

Index

A

B

C

D

E

F

I

L

N

O

P

Q

R

S

T