Migrating from EDMA v2.0 to EDMA v3.0 for TMS320TCI648x DSP

Xiaohui Li *DSP Software Applications*

1 Introduction

The TMS320TCI648x devices introduce a newly designed Enhanced Direct Memory Access (EDMA3). The EDMA3 has many new features that improve system performance and enhance debugging capabilities. This document summarizes the key differences between EDMA3 on the TCI648x devices and EDMA2 on the TMS320C64x devices. This document also provides guidance for migrating from EDMA2 to EDMA3. For more detailed information about EDMA3, see the *TMS320TCI648x DSP Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (SPRU727). For more detailed information about EDMA2, refer to the *TMS3206000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide* (SPRU234).

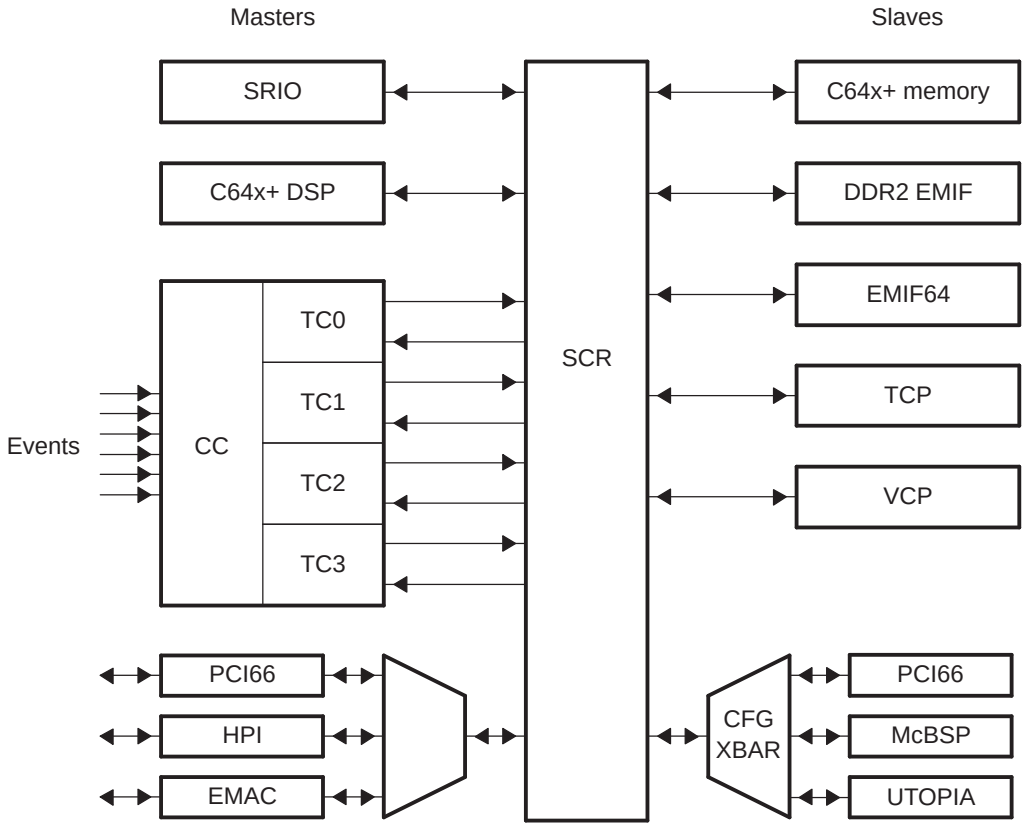


Figure 1. TMS320TCI648x Device Block Diagram

1.1 System Architecture Overview

Figure 1 shows the high-level system architecture of a TMS320TCI648x device.

The TCI648x device consists of masters, slaves, and a Switched Central Resource (SCR), as shown in Figure 1. The masters and slaves are cross connected through the SCR. Masters initiate data transfer between masters and slaves. Masters include the C64x+ CPU (DSP), Rapid I/O interface, four EDMA Transfer Controllers (TCs), and a crossbar port connecting three master peripherals (PCI, HPI, and EMAC) to the SCR. The slaves include C64x+ DSP memory, DDR EMIF, Turbo Co-processor (TCP), Viterbi Co-processor (VCP), and a crossbar connecting many peripherals to the SCR.

The SCR provides connectivity and arbitrations among masters and slaves. It allows truly concurrent data transfers between unique connections between masters and slaves. For example, the connection between PCI to DDR EMIF is independent of the connection between McBSP and DSP L2. The two data transfers are in complete parallel. SCR handles arbitration when multiple masters access the same slave. You can program the priority levels among the masters.

There is a key difference between the TCI648x device-based system architecture and the C64x DSP-based system architecture. On TCI648x devices, the EDMA3 transfer controller and other master peripherals such as EMAC, HPI, and PCI on the TCI648x device are all masters. EDMA3 only handles the data transfer between slaves. The transfers between other master peripherals and slaves do not go through EDMA.

Both regular EDMA transfers and transfers between master peripherals and slaves go through the EDMA2 transfer controller on C64x devices. Thus, all of the EDMA transfers and other master peripheral transfers compete for the bandwidth sources of the EDMA2 transfer controller.

The system architecture of the TCI648x devices offers increased system performance and is more concurrent.

1.2 EDMA3 Overview

The newly introduced EDMA3 on the TCI648x devices consists of a Channel Controller (CC) and four Transfer Controllers (TC). See Figure 2. You can program EDMA3 for data transfer between two slaves of a Switched Central Resource (SCR). Programmed transfers are typically between a slave peripheral and a memory (e.g., McBSP to/from L2 SRAM) or between two slave memories (e.g., L2 SRAM to/from EMIF). EDMA3 supports 64 DMA channels and 4 QDMA channels on the TCI648x device.

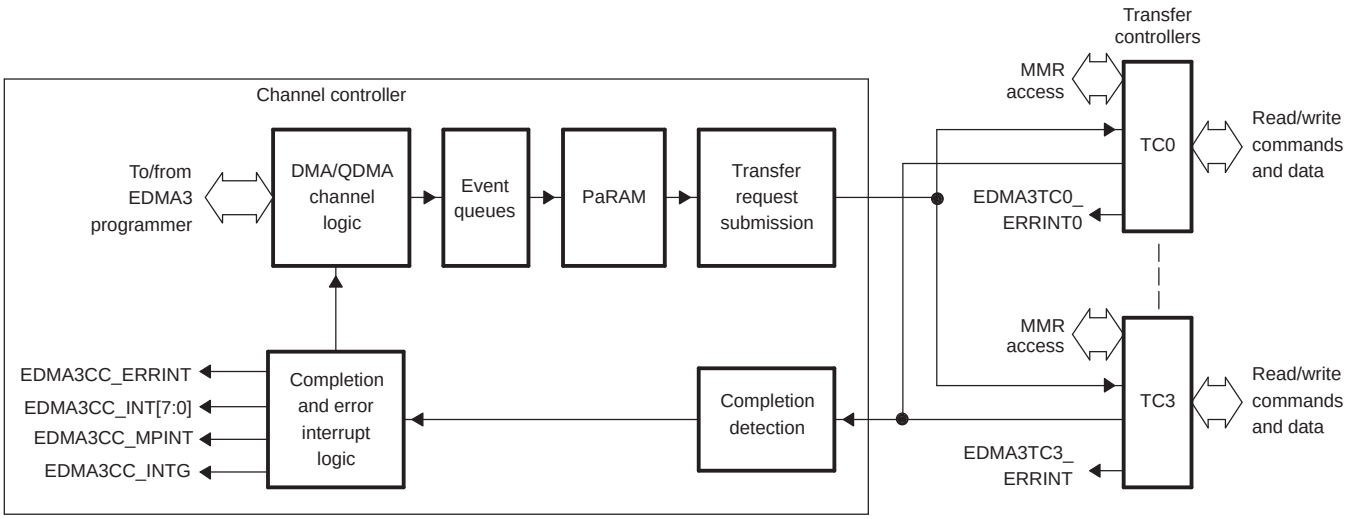


Figure 2. EDMA3 Block Diagram

1.2.1 EDMA3 Channel Controller Overview

Figure 3 shows a high-level block diagram for the EDMA3 Channel Controller (CC). The CC serves as the user interface for EDMA3. The CC provides the functionality to detect trigger events, prioritize DMA events, queue, and submit transfer requests to the EDMA Transfer Controller (TC) for data movement between slave peripherals.

EDMA3 supports two types of DMA channels: regular DMA channels and QDMA channels. The following can trigger DMA channels:

- External events (e.g., the McBSP transmit event and receive event)
- Software writing a 1 to a given bit location of the event set register of a channel
- Chaining

The CC arbitrates among all pending DMA/QDMA events with a fixed 64:1 and 8:1 priority encoder for DMA and QDMA events, respectively (the lowest channel number is the highest priority). DMA events always have higher priority than QDMA events. The winning event is queued into the event queue. There are four event queues. Each event queue can have a maximum of 16 queued events. Events are dequeued in FIFO order. If more than one TC is ready for programming a transfer request, the event queues are serviced with fixed priority: Q0 is always the highest and Q3 is always the lowest. It is possible to designate one word of the PaRAM set to be the trigger word. When the CPU writes to this trigger word, the QDMA channel is triggered.

The CC contains 256 parameter RAM (PaRAM) sets that supports flexible ping-pong, circular buffering, channel chaining, auto reloading, memory protection, etc.

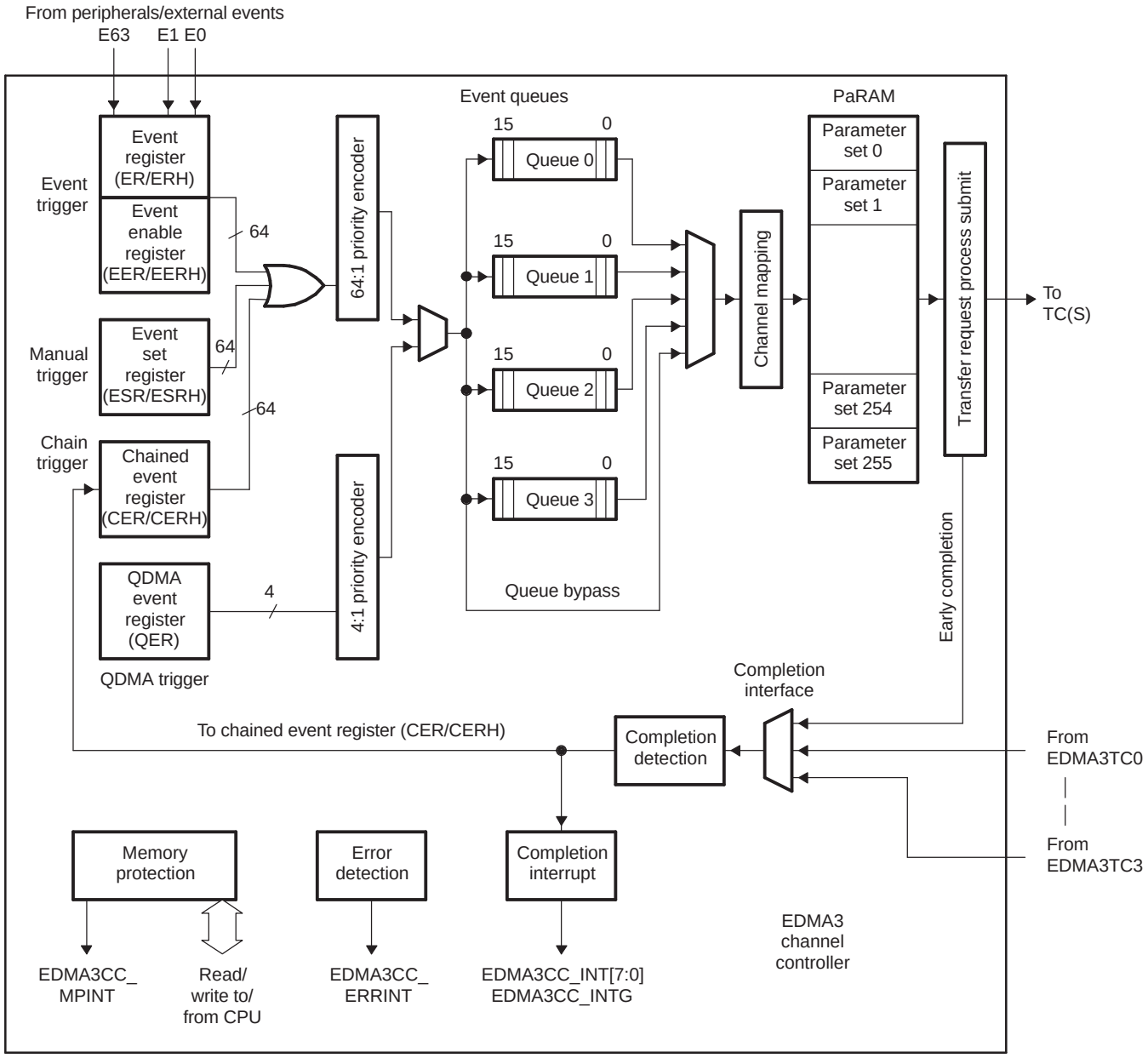


Figure 3. EDMA3 Channel Controller Block Diagram

As soon as the TC is available for programming a new transfer request into it, events are extracted from the event queue. As an event is extracted from the event queue, the associated PaRAM entry is processed and submitted to the TC. The CC updates the appropriate counts and addresses in the PaRAM entry in anticipation of the next trigger event for that PaRAM entry.

1.2.2 EDMA3 Transfer Controller Overview

The Transfer Controller's (TC's) primary responsibility is to perform read and write transfers to the slaves connected to the SCR, as the PaRAM entries specify. Figure 4 shows a high-level block diagram of the EDMA3 TC. For more information, see the *TMS320TCI648x DSP Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (SPRU727).

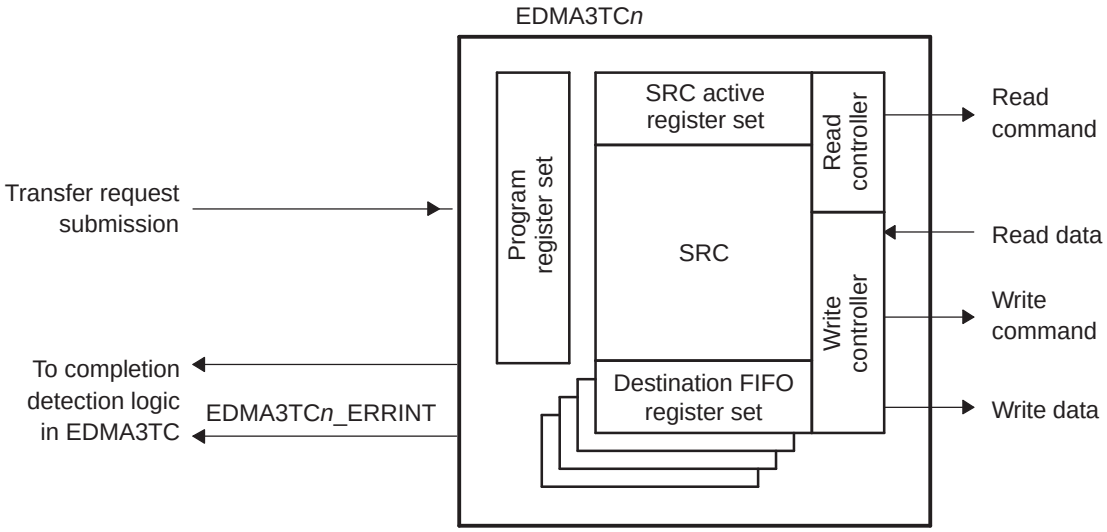


Figure 4. EDMA3 Transfer Controller Block Diagram

1.3 Architecture Comparison Between EDMA2 and EDMA3

On EDMA3-based devices, both the EDMA3 Channel Controllers (CCs) and the other master peripherals are masters of a Switched Central Resource (SCR). The EDMA3 CCs and TCs only handle data transfers between two slaves, such as McBSP to DDR and/or DDR to DSP L2 memory. All other master peripherals connect directly to the SCR. There are two levels of prioritization related to EDMA:

- The first level handles prioritization and arbitration among all of the DMA and QDMA schemes within the EDMA channel controller.
- The second level is the system level prioritization of the EDMA transfer controller versus other masters. The SCR performs this system level priority resolution.

All of the DMA and master transfers go through the EDMA Transfer Controller (TC) on EDMA2. All of the DMA transfers and other master transfers compete for the TC bandwidth. The prioritization among all of the transfers is done within the TC.

Compared to EDMA2, the EDMA3-based devices provide more bandwidth for both DMA/QDMA and other master related data transfers and are more concurrent. The EDMA3-based devices are capable of providing concurrent transfers for both DMA/QDMA and master transfers, assuming that all of the masters go to different slaves.

1.4 Overview Feature Differences and New Features on EDMA3

Table 1 summarizes the major differences between EDMA2 on C64x DSP and EDMA3 on TCI648x DSP.

Table 1. Summary of Major Differences Between EDMA2 and EDMA3

Feature	Description	EDMA2	EDMA3
PaRAM	PaRAM set size	Six 32-bit words	Eight 32-bit words
	Number of PaRAM entries	Number depends on devices	256
	Event mapping to PaRAM	Fixed	Flexible
	Indexing	Same for source and destination	Different for source and destination
Transfer Type	Dimension of transfer	Two	Three
	Synchronization types	Element synchronized 1D	A-synchronized transfer
		Frame synchronized 1D	AB-synchronized transfer
		Array synchronized 2D	
		Block synchronized 2D	
QDMA	QDMA location	Outside of EDMA	Part of EDMA
	QDMA channel mapping	N/A	Flexible
	QDMA triggering	Write to QDMA pseudo register	Write to trigger word (programmed)
	QDMA linking	N/A	Flexible
	STATIC bit in OPT for QDMA	N/A	New feature on EDMA3
Interrupt	Interrupt types	No error interrupt	Transfer completion interrupt, error interrupt
	Interrupt registers		Introduces some new interrupt registers
	Region interrupt	N/A	New feature
	Interrupt service routine		
Event Queue	Number of event queues	4	4
	Queue length	16	16
	Queue usage	Shared by DMA/QDMA and other master peripherals	Only used by EDMA/QDMA channels
	Event queue timing	Fire-and-Forget	Wait for TR to be submitted to TC to update PaRAM
Priority Processing	Mapping between events and event queues	Through PRI of OPT field	Through queue number registers
	Queue priority versus other master transfers	Arbitrated at Transfer Controller (TC)	Arbitrated at Switch Center Resource (SCR)
Linking	DMA linking feature availability	Available, optional	Always
	QDMA linking feature availability	N/A	Always
	Terminating linking	Link to a NULL entry	Configure FFFFh of the LINK field of the last linked PaRAM
Chaining	Chaining comparison	Use the same bits for enabling chain and interrupt	Use different bits for enabling chain and interrupt
Set and Clear	Set and clear operation	N/A	New feature on EDMA3
DMA Region Access	DMA region access	N/A	New feature on EDMA3
Debug Visibility	Event queue visibility	N/A	New feature on EDMA3
	Event queue water marking	N/A	New feature on EDMA3
	Event queue threshold	N/A	New feature on EDMA3
Error Detection	Event missed registers	N/A	New feature on EDMA3
	Channel Controller error registers	N/A	New feature on EDMA3
	Error from null parameter entry	N/A	New feature on EDMA3
	Error interrupts	N/A	New feature on EDMA3

2 Comparison Between EDMA2 and EDMA3

2.1 PaRAM Set

The EDMA3 controller is a RAM-based architecture like EDMA2. Each EDMA channel transfer context (source/destination addresses, count, indexes, etc.) is programmed in a parameter table (see Figure 5 and Figure 6). The table is a block of Parameter RAM (PaRAM) that is located within the EDMA Channel Controller (CC). Although both EDMA2 and EDMA3 use RAM-based architecture, there are key differences related to PaRAM entries. This section provides a summary of the differences.

	31	0
Word 0	Channel Options Parameter (OPT)	
Word 1	Channel Source Address (SRC)	
Word 2	Array/Frame Count (FRMCNT)	Element Count (ELECNT)
Word 3	Channel Destination Address (DST)	
Word 4	Array/Frame Index (FRMIDX)	Element Index (ELEIDX)
Word 5	Element Count Reload (ELERLD)	Link Address (LINK)

Figure 5. PaRAM Entry for EDMA2

	31	0
Word 0	Channel Options Parameter (OPT)	
Word 1	Channel Source Address (SRC)	
Word 2	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Word 3	Channel Destination Address (DST)	
Word 4	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
Word 5	BCNT Reload (BCNTRLD)	Link Address (LINK)
Word 6	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
Word 7	Reserved	Count for 3rd Dimension (CCNT)

Figure 6. PaRAM Entry for EDMA3

2.1.1 PaRAM Entries

Figure 5 and Figure 6 show the definitions of PaRAM entries for both EDMA2 and EDMA3. The PaRAM set size is six 32-bit words on EDMA2 (Figure 5).The PaRAM set size is eight 32-bit words on EDMA3 (Figure 6). Table 2 summarizes the comparisons between the two PaRAM entries.

Table 2. Summary of PaRAM Comparison Between EDMA2 and EDMA3

PaRAM Entry	EDMA2	EDMA3
OPT field	Refer to the *TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide* (SPRU234).	Introduces significant changes. See the *TMS320TCI648x DSP Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (SPRU727).
Count	Only ELECNT and FRMCNT are available to specify two-dimension transfers.	Introduces ACNT, BCNT, and CCNT to specify the sizes for three-dimension transfers.
Index	Both source and destination use the same indexes: ELEIDX and FRMIDX.	Introduces source indexes (SRCBIDX and SRCCIDX) and destination indexes (DSTBIDX and DSTCIDX) and allows independent source and destination indexing.
DST and SRC fields	Both EDMA2 and EDMA3 have 32-bit source and destination fields that specify the source and destination addresses.	
Link field	Both EDMA2 and EDMA3 have a 16-bit link field that specifies the linking address.	
Reload field	The ELERLD field on EDMA2 is equivalent to the BCNTRLD field on EDMA3	

2.1.2 Number of PaRAM Entries

Table 3 summarizes the number of PaRAM entries for some EDMA2-based devices and EDMA3-based devices. Refer to the device-specific data manual for the number of PaRAM entries on the device.

Table 3. Number of PaRAM Entries for Different Devices

Device	EDMA Type	Number of PaRAM Entries
TMS320C6415 DSP	EDMA2	83
TMS320DM642 DSP	EDMA2	200
TMS320TCI648x DSP	EDMA3	256

2.1.3 Event Mapping to PaRAM

The 64 DMA channels and the 8 QDMA channels are flexibly mappable to any of the 256 available PaRAM entries on EDMA3 on TCI648x devices, as shown in Figure 7. Any of the 64 DMA channels are mappable to any of the 256 PaRAM entries through DMA channel mapping registers DCHMAP0 to DCHMAP63. Any of the 4 QDMA channels are mappable to any of the available 256 PaRAM entries through QDMA channel mapping registers QCHMAP0 to QCHMAP3.

Comparatively, on EDMA2, the 64 DMA channels are the only channels that map to the PaRAM entries. The mappings between DMA events and PaRAM entries are fixed. For example, event 0 is permanently mapped to the first PaRAM entry and event 1 is permanently mapped to the second PaRAM, etc. Five QDMA registers (not PaRAM) are used to configure the QDMA transfers.

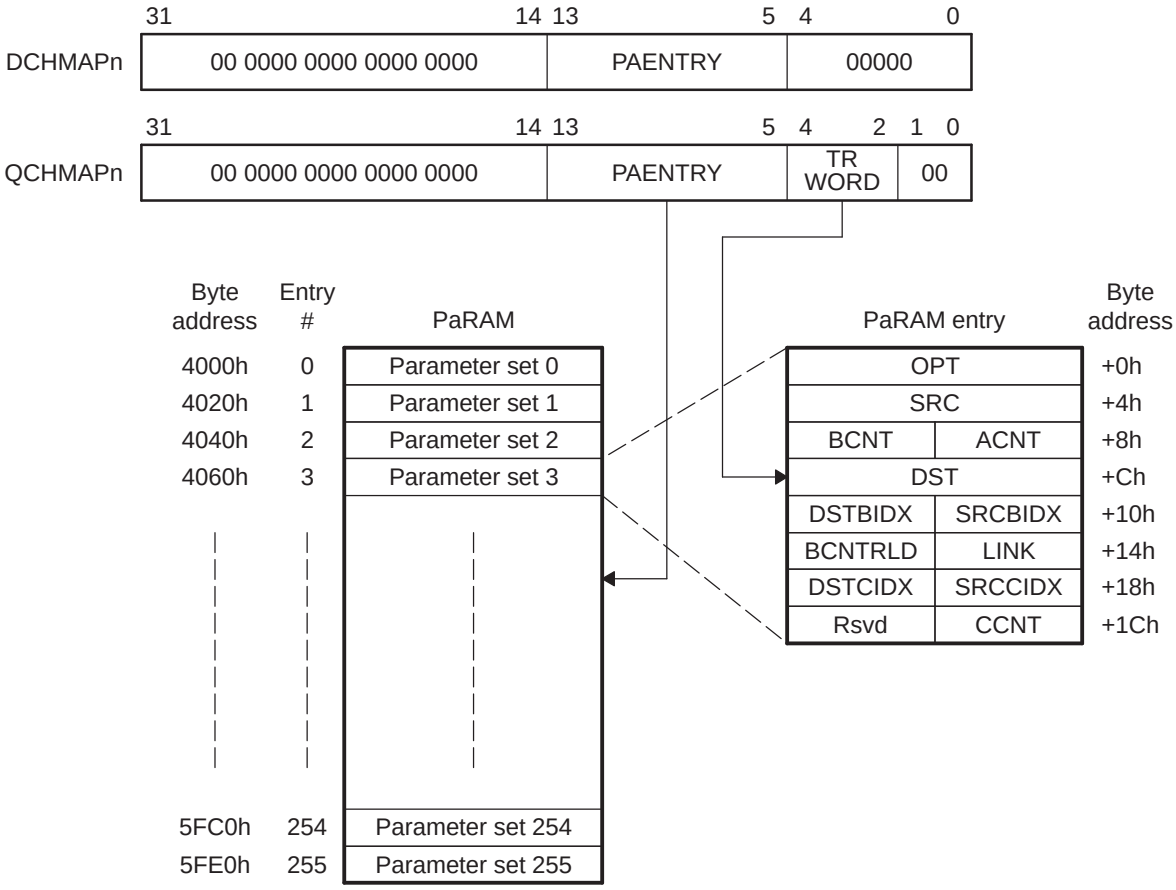


Figure 7. DMA and QDMA Channel Mapping to PaRAM

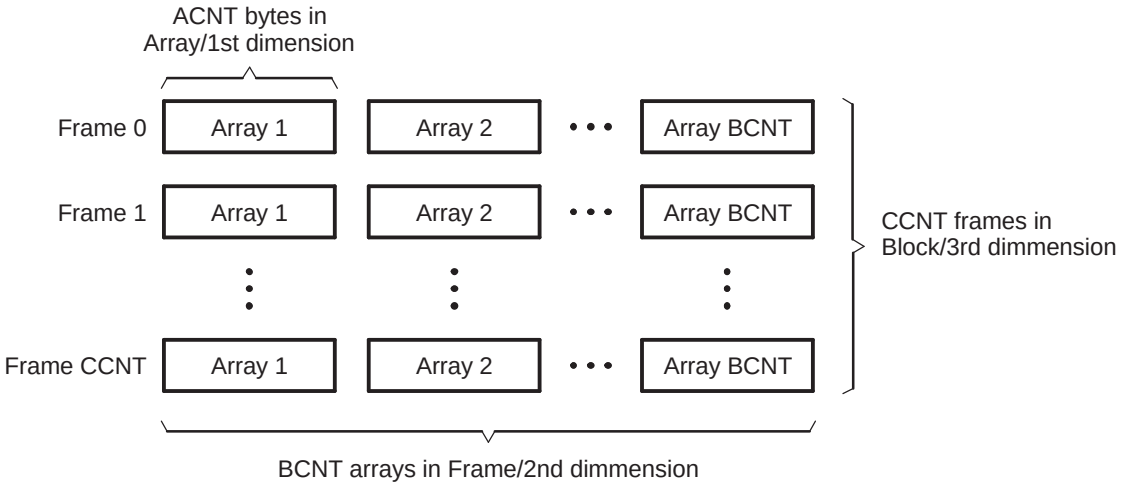


Figure 8. Dimension of Array, Frame, and Block

2.1.4 Dimension of Transfer

Unlike EDMA2, EDMA3 uses a more orthogonal structure for its DMA transfers. Three dimensions define each transfer. Of the three dimensions, EDMA3 only supports two synchronization types: A-synchronized transfers and AB-synchronized transfers. Chaining allows the logical achievement of ABC-synchronized transfers. The three dimensions are shown in Figure 8. A summary of the three dimensions follows:

- 1st Dimension or Array (A): The first dimension in a transfer consists of ACNT contiguous bytes. The total transfer size is ACNT bytes.
- 2nd Dimension or Frame (B): The second dimension in a transfer consists of BCNT arrays of ACNT bytes. Each array transfer in the second dimension is separated from the other array transfers by using SBIDX or DBIDX to program an index. The total transfer size is ACNT × BCNT bytes.
- 3rd Dimension or Block (C): The third dimension in a transfer consists of CCNT frames of BCNT arrays programmed using SCIDX or DCIDX. The total transfer size is ACNT × BCNT × CCNT bytes.

The following defines the transfer on EDMA2:

- 1st Dimension or Frame: The first dimension in a transfer consists of ELECNT elements and ELEIDX separates the consecutive elements. The element size is defined as ELECNT × ESIZE bytes, if ESIZE is configured as number of bytes.
- 2nd Dimension: The second dimension in a transfer consists of FRMCNT frames of ELECNT elements. Each frame transfer in the second dimension is separated from the other frame transfers by the FRMIDX index. The total transfer size is FRMCNT × ELECNT × ESIZE bytes, if ESIZE is configured as number of bytes.

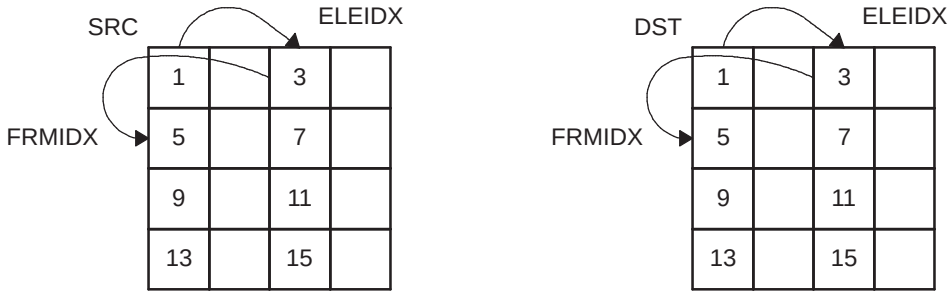


Figure 9. Indexing Example for EDMA2

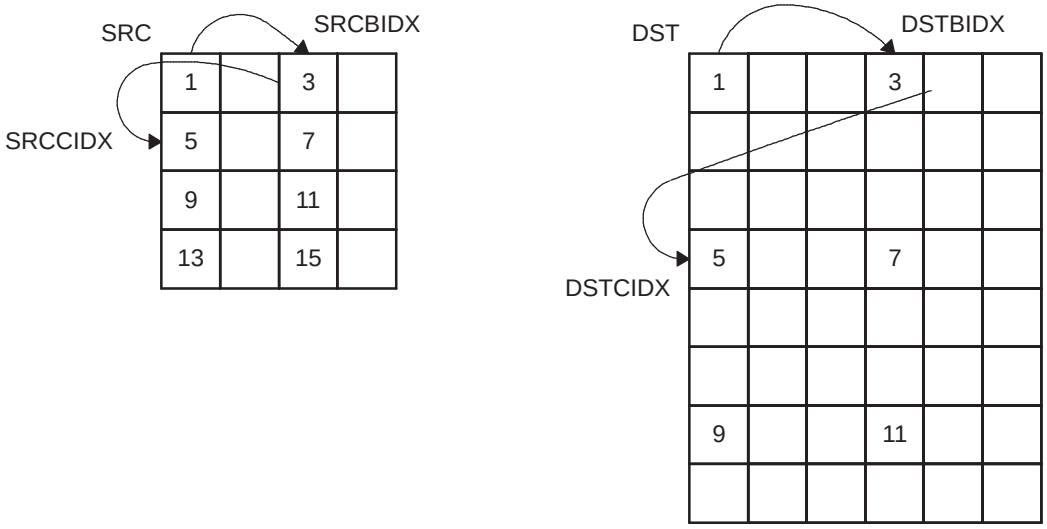


Figure 10. Indexing Example for EDMA3

2.1.5 Indexing

EDMA2 uses the same two indexes (ELEIDX and FRMIDX) to specify the offset between the two consecutive elements or the two frames for both the source and for the destination.

EDMA3 uses different indexes for the source and for the destination. EDMA3 uses SRCBIDX and DSTBIDX to specify the offsets between the two consecutive arrays and two consecutive frames for source indexing. EDMA3 uses DSTBIDX and DSTCIDX to specify destination indexing. Thus, the source and destination can use completely different indexing on EDMA3. Figure 9 and Figure 10 show two examples that demonstrate the potential indexing differences between EDMA2 and EDMA3.

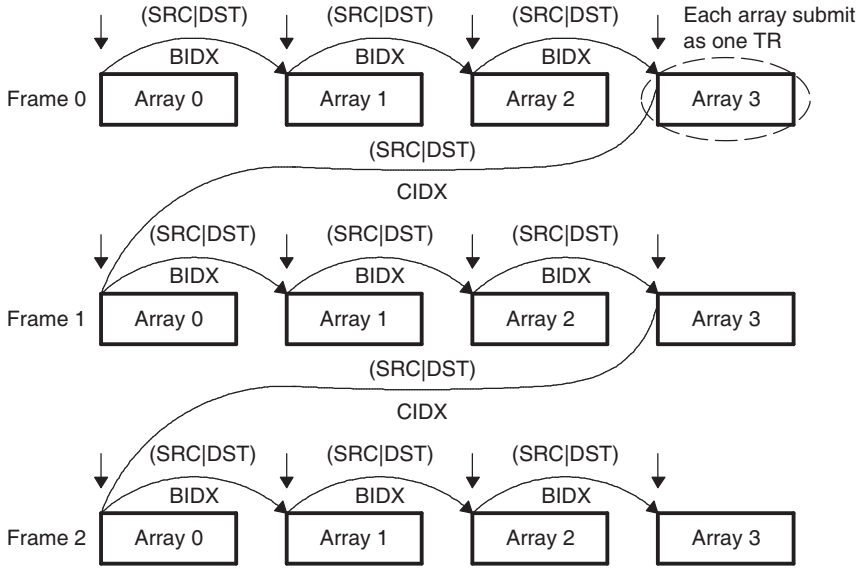


Figure 11. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)

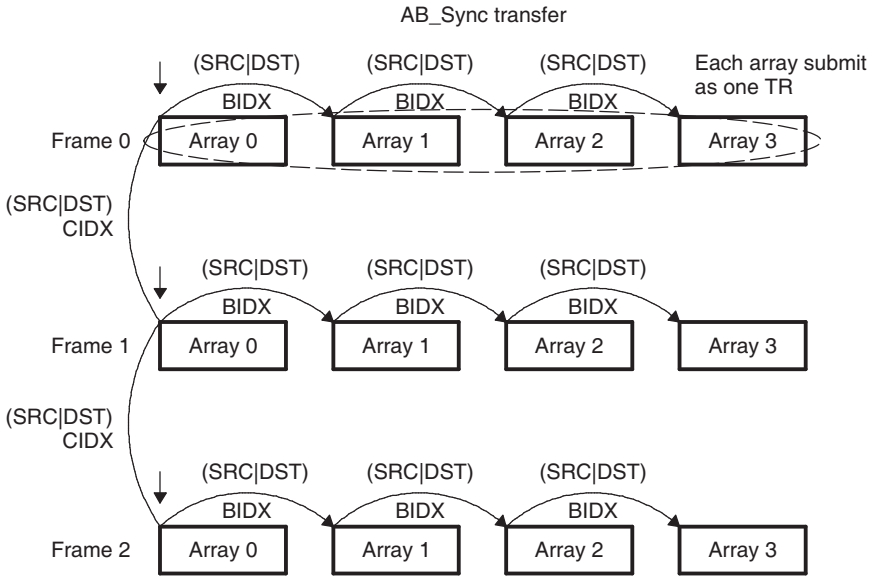


Figure 12. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)

2.2 EDMA3 Synchronization Type

There are two synchronization types on EDMA3: A and AB.

- *A-Synchronized Transfers:* In an A-synchronized transfer, each transfer request sync triggers the

transfer of the first dimension of ACNT bytes, or one array of ACNT bytes. Thus, each TR packet consists of transfer information for only one array. SRCBIDX and DSTBIDX can separate arrays where the start address of Array N is equal to the start address of Array N – 1 + (SRCBIDX or DSTBIDX), as shown in Figure 11,

- *AB-Synchronized Transfers*: In an AB-synchronized transfer, each transfer request sync triggers the transfer for two dimensions, or for one entire frame of BCNT of ACNT bytes. SRCBIDX and DSTBIDX can separate arrays. SRCIDX and DSTIDX can separate frames. After submitting a transfer request, the address updates are performed by adding SRCIDX and DSTIDX to the beginning address of the beginning array in the frame, as shown in Figure 12.

EDMA2 has the following synchronization types for its DMA transfers:

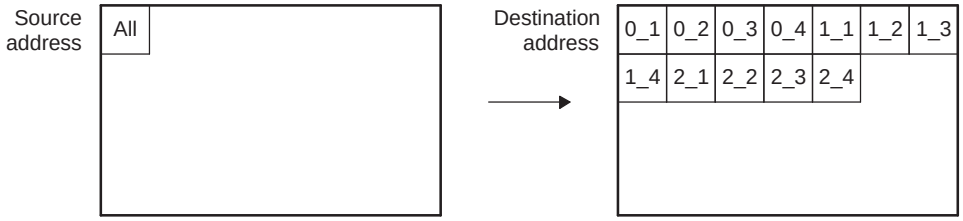
- Element Synchronized 1D Transfer (FS = 0)
- Frame Synchronized 1D Transfer (FS = 1)
- Array Synchronized 2D Transfer (FS = 0)
- Block Synchronized 2D Transfer (FS = 1)

For more information on EDMA2 synchronization types, refer to the *TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide* (SPRU234).

2.3 Example of Element Synchronized 1D-to-1D Transfer on EDMA2

Figure 13 shows a block diagram of an element synchronized 1D-to-1D transfer and the configuration on EDMA2. The source address is from a fixed location. The data is copied to continuous addresses in the destination.

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter	
2020 0000h		Channel Options Parameter (OPT)	
Source Address		Channel Source Address (SRC)	
0002h	0004h	Array/Frame Count (FRMCNT)	Element Count (ELECNT)
Destination Address		Channel Destination Address (DST)	
Don't care	Don't care	Array/Frame Index (FRMIDX)	Element Index (ELEIDX)
0004h	Don't care	Element Count Reload (ELERLD)	Link Address (LINK)

(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
Reserved	TCCM	Reserved	LINK	FS

Figure 13. Element Synchronized 1D-to-1D Transfer (SUM = 00, DUM = 01) on EDMA2

The EDMA2 element synchronized 1D-to1D transfer uses the A-synchronization type on EDMA3. Every synchronization event transfers ACNT bytes from the source to the destination until it exhausts all of the counters. Figure 14 shows the configuration for PaRAM entry and the Channel Options Parameter (OPT) content for this transfer example.

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 000Bh		Channel Options Parameter (OPT)	
Source Address		Channel Source Address (SRC)	
0004h	0004h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Destination Address		Channel Destination Address (DST)	
0004h	0004h	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0004h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0004h	0004h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0003h	Reserved	Count for 3rd Dimension (CCNT)

(b) EDMA Channel Options Parameter (OPT) Content

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	xx	00
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

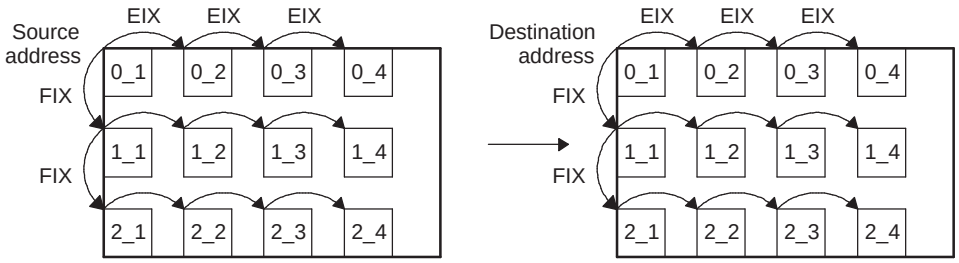
15 12	11	10 8	7 4	3	2	1	0
0000	0	000	0000	1	0	1	1
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

Figure 14. Element Synchronized 1D-to-1D Transfer on EDMA3

2.4 Example of Frame Synchronized 1D-to-1D Transfer on EDMA2 and EDMA3

Figure 15 shows a block diagram of a frame synchronized 1D-to-1D transfer and the configuration on EDMA2. Every synchronization event transfers a frame of elements from the source location to the destination.

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents	
2360 0001h	
Source Address	
0002h	0004h
Destination Address	
FIX (frame index)	EIX (element index)
Don't care	Don't care

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Array/Frame Count (FRMCNT)	Element Count (ELECNT)
Channel Destination Address (DST)	
Array/Frame Index (FRMIDX)	Element Index (ELEIDX)
Element Count Reload (ELERLD)	Link Address (LINK)

(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
Reserved	TCCM	Reserved	LINK	FS

Figure 15. Frame Synchronized 1D-to-1D Transfer (SUM = 11, DUM = 11) on EDMA2

This EDMA2 frame synchronized 1D-to-1D transfer uses the AB-synchronization type on EDMA3. Every synchronization event transfers ACNT × BCNT bytes from the source to the destination until it exhausts all of the counters. Figure 16 shows the configuration for PaRAM entry and the Channel Options Parameter (OPT) content for this transfer example.

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 000Fh		Channel Options Parameter (OPT)	
Source Address		Channel Source Address (SRC)	
0004h	0004h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Destination Address		Channel Destination Address (DST)	
EIX (element index)	EIX (element index)	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
FIX (frame index)	FIX (frame index)	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0003h	Reserved	Count for 3rd Dimension (CCNT)

(b) EDMA Channel Options Parameter (OPT) Content

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	xx	0
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

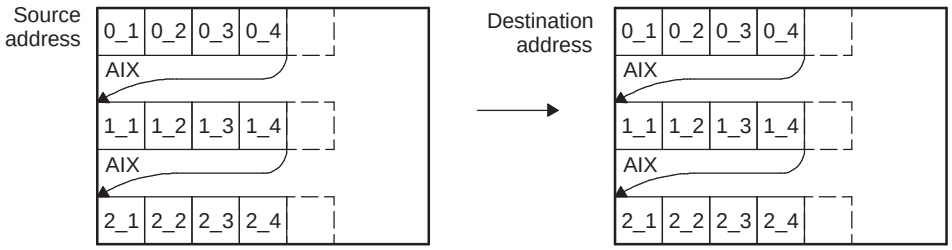
15 12	11	10 8	7 4	3	2	1	0
0	0	000	0000	1	1	1	1
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

Figure 16. Frame Synchronized 1D-to-1D Transfer on EDMA3

2.5 Example of Block Synchronized 2D-to-2D Transfer on EDMA2 and EDMA3

Figure 17 shows a block diagram of a block synchronized 2D-to-2D transfer and the configuration on EDMA2. Every synchronization event copies a block of several frames of data from the source address to the destination address.

(a) Transfer Path



(b) EDMA Parameters

Parameter Contents		Parameter	
65A0 0001h		Channel Options Parameter (OPT)	
Source Address		Channel Source Address (SRC)	
0002h	0004h	Array/Frame Count (FRMCNT)	Element Count (ELECNT)
Destination Address		Channel Destination Address (DST)	
AIX (array index)	Don't care	Array/Frame Index (FRMIDX)	Element Index (ELEIDX)
Don't care	Don't care	Element Count Reload (ELERLD)	Link Address (LINK)

(C) EDMA Channel Options Parameter (OPT) Content

31 29	28 27	26	25 24	23	22 21	20	19 16
011	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
Reserved	TCCM	Reserved	LINK	FS

Figure 17. Block Synchronized 2D-to-2D Transfer (SUM = 01, DUM = 01) on EDMA2

The EDMA2 block synchronized 2D-to-2D transfer uses the AB-synchronization type on EDMA3. The ACNT × BCNT × CCNT bytes specify a block of data that is transferred from the source to the destination at every synchronization event. Figure 18 shows the configuration for both the PaRAM entry table and the Channel Options Parameter (OPT) field for this AB-synchronized 2d transfer.

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 000Fh		Channel Options Parameter (OPT)	
Source Address		Channel Source Address (SRC)	
0003h	0010h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Destination Address		Channel Destination Address (DST)	
10h + AIX (array index)	10h + AIX (array index)	Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) EDMA Channel Options Parameter (OPT) Content

31	30 28	27 24	23	22	21	20	19 18	17 16
0	000	0000	0	0	0	1	0	0
PRIV	Reserved	PRIVID	ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	Reserved	TCC

15 12	11	10 8	7 4	3	2	1	0
0	0	000	0000	1	1	1	1
TCC	TCCMOD	FWID	Reserved	STATIC	SYNCDIM	DAM	SAM

Figure 18. Block Synchronized 2D-to-2D Transfer on EDMA3

2.6 QDMA Comparison Between EDMA2 and EDMA3

2.6.1 QDMA Architecture Differences Between EDMA2 and EDMA3

Table 4 summarizes the differences of the QDMA architecture between EDMA2 and EDMA3.

Table 4. Summary of QDMA Architecture Differences Between EDMA2 and EDMA3

Feature	EDMA2	EDMA3
QDMA location	QDMA is part of the L2 cache controller that is a separate hardware module from EDMA channel and transfer controllers.	QDMA is an integral part of the EDMA channel controller, as shown in Figure 3.
PaRAM versus register for QDMA context saving	QDMA registers and pseudo registers are used for storing the contexts for its QDMA transfers.	Similar to DMA transfer, the PaRAM entry table is used for storing the QDMA transfer context, including synchronization type, source and destination addresses, etc.
QDMA event queue comparison	Event queue stores the actual transfer context from either PaRAM for DMA or from QDMA registers for QDMA transfer once the transfer request is admitted into the event queue.	The event queues are used for storing the indexes of DMA and QDMA events, rather than the actual context of PaRAM entries of the transfers. The actual transfer context is only loaded into the transfer when the transfer request is submitted to the EDMA transfer controller.
	The difference between EDMA2 and EDMA3 is that on EDMA2, once the event queue admits the transfer request, the QDMA registers are ready for reconfiguration for a new transfer. However, on EDMA3, you must wait until the transfer request is submitted into the transfer controller to begin configuring the related PaRAM entry for a new transfer, even after programming the QDMA PaRAM set. Otherwise, the content of the PaRAM entry of the current QDMA transfer may become corrupt while the transfer request is waiting in the event queue. You must be aware of this difference when configuring PaRAM for QDMA transfer on EDMA3.	
Number of QDMA channels	There is only one QDMA channel. EDMA2 can have multiple outstanding QDMA transfer requests waiting in the event queue because the event queue on EDMA2 stores the actual transfer context. Once the current QDMA transfer request is admitted into the queue, you can submit a new QDMA transfer. It is equivalent to having multiple QDMA channels. You can allocate up to a maximum 7 queue length of an event queue for QDMA on EDMA2. Equivalently, it has up to 7 QDMA channels.	Four QDMA channels. You can submit up to four outstanding QDMA requests on TCI648x DSP at any time.

2.6.2 QDMA Channel Mapping to PaRAM on EDMA3

Similar to DMA transfer, EDMA3 uses the PaRAM entry table to specify the QDMA transfer. As previously mentioned, each PaRAM entry has eight 32-bit words. The PAENTRY field in the QDMA channel mapping register (QCHMAP*n*) is used for mapping QDMA channel *n* to any one of the PaRAM entries, as shown in Figure 7.

As mentioned in the previous section, on EDMA2, QDMA does not use the PaRAM for storing transfer context. Two sets of registers (QDMA and QDMA pseudo registers) are used for storing configuration information for QDMA transfer instead.

2.6.3 Triggering QDMA on EDMA3

You can specify a trigger word from any one of the eight 32-bit words of PaRAM for QDMA on EDMA3. Writing to the trigger word triggers the Channel Controller (CC) of QDMA to issue a transfer request. The trigger word (TRWORD) field in the QDMA Channel Mapping Register (QCHMAP*n*) defines the trigger word for a particular QDMA channel *n*, as shown in Figure 7.

This flexibility enables the CPU to selectively modify only the PaRAM entry that requires modification, thereby triggering the transfer. For example, after a transfer, if only the count must change, then you can configure QCHMAP*n* so that the count is the trigger word and a write to it automatically triggers the transfer.

Writing to the QDMA pseudo registers leads to submitting a QDMA transfer request on EDMA2.

2.6.4 Linking on QDMA

Linking is always enabled for both DMA and QDMA transfers on EDMA3. Linking is not available for QDMA and the LINK address field in the Channel Options Parameter (OPT) is reserved on EDMA2. Refer to Section 2.10 for a more detailed comparison between EDMA2 and EDMA3 linking.

When a QDMA transfer completes, its PaRAM updates from its linked PaRAM entry because linking is always enabled on EDMA3. You are responsible for configuring the termination of the linked operation for any enabled QDMA operation. You must set the LINK address field to FFFFh. You must also set the STATIC bit to 1 in OPT of the last PaRAM entry to terminate a QDMA linked list.

2.6.5 Using IDMA for Efficient Configuration of QDMA PaRAM Sets on EDMA3

The Internal DMA (IDMA) is a newly introduced DMA on the TCI648x devices and is part of the TMS320C64x+ DSP megamodule. IDMA performs fast block transfers between two linear ranges of memory locations local to the device's DSP megamodule. IDMA has a mask register that allows you to enable/disable transfers to specific words within the linear range. The local memory includes Level 1 Program (L1P), Level 1 Data (L1D), Level 2 (L2) Memories, and peripheral configuration registers. For more information on IDMA, refer to the *TMS320C64x+ DSP Megamodule Reference Guide* (SPRU871).

IDMA is used to efficiently configure the PaRAM for QDMA operations. For example, if the last word of PaRAM set is configured as the trigger word, then you can use IDMA to copy the entire content to the PaRAM set. The QDMA Event Register (QER) sets and submits a transfer request when IDMA finishes transferring the QDMA.

IDMA is also used to efficiently configure multiple PaRAM entries for linking applications. For example, you can allocate consecutive PaRAM entries for a particular QDMA in an application that requires linking multiple PaRAM entries for QDMA operation. With the last PaRAM entry (at the highest memory location) used for the first QDMA transfer, you can use IDMA to write all of the PaRAM entries allocated for the QDMA. Completion of the transfer of data to the last PaRAM entry triggers QDMA and linking to the rest of PaRAM happens automatically.

2.6.6 STATIC Bit in OPT and QDMA on EDMA3

EDMA3 introduces the STATIC bit in the Channel Options Parameter (OPT).

When STATIC = 0, PaRAM entries update during the course of the transfer. The link updates when a QDMA transfer completes. When STATIC = 1, none of the PaRAM entries update during a transfer. Linking is disabled when the transfer completes.

EDMA3 always enables linking. You must specify a valid address of a PaRAM entry table in the LINK field of each of its linked PaRAM entries to form a linked list. Configure the STATIC bit to 0 to allow PaRAM fields to update for all of the linked PaRAM entries other than the last. You must set the STATIC bit and configure the LINK field to FFFh in the last linked PaRAM entry of QDMA transfers to terminate a QDMA operation.

If a particular transfer only requires one QDMA transfer, then configure the LINK field to FFFFh and set the STATIC bit to 1 for the PaRAM set.

2.6.7 Software Use Model on EDMA3

When an event queue of the Channel Controller (CC) admits a transfer request from either DMA or QDMA, EDMA3CC only queues the event number, not the actual PaRAM entry.

Only after the associated PaRAM set is submitted to the transfer controller, you can configure the PaRAM entry for a new QDMA transfer.

It is important to note that when you are reprogramming the PaRAM entry of a QDMA, you must check to ensure that the last linked QDMA event has been submitted to EDMA TC. Otherwise, the contents of the PaRAM entry of a QDMA transfer request may become corrupt while the request is still waiting in the event queue.

There is an issue when you know that a linking operation of QDMA is complete and the associated PaRAM entry is used for new QDMA operations. Figure 19 shows an example of a linking operation and how the values of the QDMA Event Register (QER) and the QDMA Secondary Event Register (QSER) change throughout the entire linking operation. See the later part of this document for more information about the secondary event registers.

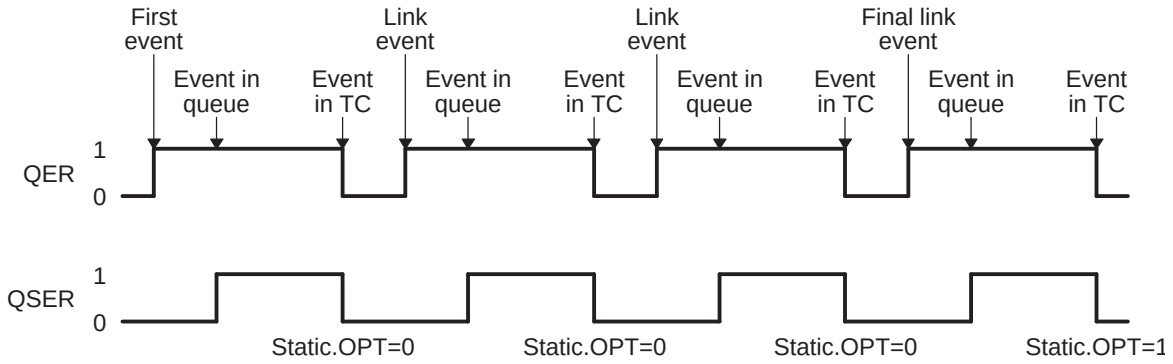


Figure 19. Linking Example with QER and QSER Values

There are two methods for tracking the status of PaRAM resources for any DMA and QDMA operation, as summarized below:

- Wait for the interrupt to complete. You can wait for a particular DMA and/or QDMA transfer to complete to determine if the PaRAM is available to use for new data transfers.
- Poll for transfer request submission to the transfer controller. This method is particularly useful for QDMA linking operations. You can poll the status of the related bit in SER/ER for DMA or QSER/QER for QDMA and the STATIC bit in OPT to determine if the related transfer request is submitted to the transfer controller. If both SER/ER or QSER/QER bits are zeros and the STATIC bit equals 1, then the last linked QDMA event has been submitted to the transfer controller. The related PaRAM is also available to configure for future use.

Once a QDMA linking operation is started, the QDMA channel will not be available until after QSER:QER values become 0:0 and the value of the STATIC bit equals 1, which indicates that the last QDMA transfer is already submitted to the transfer controller. See Figure 19.

2.7 Interrupt on EDMA3

On the high level, the interrupt operation on EDMA3 is similar to that on EDMA2, but there are some significant differences. This section provides a summary of the differences.

2.7.1 Interrupt Types

EDMA3 interrupts can be divided into 2 broad categories:

- Transfer completion interrupts
- Error interrupts

The transfer completion interrupts are listed in Table 5 for TCI648x devices. The error interrupts are listed in Table 6 for TCI648x devices. For more information about the DSP Interrupt Controller (DSPINTC), refer to the *DSP Subsystem Reference Guide*. Table 5 mentions shadow region 0, 1, and 2. Refer to Section 2.7.3 for the concept of shadow region and shadow region interrupt.

Table 5. EDMA3 Transfer Completion Interrupts

Name	Description	Interrupt Number
EDMA3CC_INT0	EDMA3CC Transfer Completion Interrupt Shadow Region 0	71
EDMA3CC_INT1	EDMA3CC Transfer Completion Interrupt Shadow Region 1	72
EDMA3CC_INT2	EDMA3CC Transfer Completion Interrupt Shadow Region 2	73
EDMA3CC_INT3	EDMA3CC Transfer Completion Interrupt Shadow Region 3	74
EDMA3CC_INT4	EDMA3CC Transfer Completion Interrupt Shadow Region 4	75
EDMA3CC_INT5	EDMA3CC Transfer Completion Interrupt Shadow Region 5	76
EDMA3CC_INT6	EDMA3CC Transfer Completion Interrupt Shadow Region 6	77
EDMA3CC_INT7	EDMA3CC Transfer Completion Interrupt Shadow Region 7	78
EDMA3CC_INTG	EDMA3CC Global Transfer Completion Interrupt	24

Table 6. EDMA3 Error Interrupts

Name	Description	Interrupt Number
EDMA3CC_ERRINT	EDMA3CC Error Interrupt	79
EDMA3CC_MPINT	EDMA3CC Memory Protection Interrupt	80
EDMA3TC0_ERRINT	TC0 Error Interrupt	81
EDMA3TC1_ERRINT	TC1 Error Interrupt	82
EDMA3TC2_ERRINT	TC2 Error Interrupt	83
EDMA3TC3_ERRINT	TC3 Error Interrupt	84

2.7.2 Comparison of Interrupt Registers Between EDMA2 and EDMA3

There are equivalent parameters in the Channel Options Parameter (OPT) and registers for interrupt operation in both EDMA2 and EDMA3. Additionally, EDMA3 introduces some new interrupt registers. Table 7 summarizes the equivalent registers and parameters in OPT for interrupt operation on both EDMA2 and EDMA3.

Table 7. Equivalent Interrupt Parameters of OPT Field and Registers on EDMA2 and EDMA3

Interrupt Parameter	EDMA2	EDMA3
Transfer complete enable field in OPT	TCINT	TCINTEN
Transfer complete code field in OPT	TCCM:TCC	TCC
Intermediate transfer completion interrupt enable register	ATCINT	ITCINTEN
Intermediate transfer complete code register	ATCC	TCC
Interrupt pending registers	CIPRL, CIPRH	IPR, IPRH
Interrupt enable registers	CIERL, CIERH	IER, IERH

Table 8 summarizes the new registers EDMA3 introduces for interrupt operation.

Table 8. New Pseudo-Interrupt Registers Introduced on EDMA3

Description	Name
Interrupt clear registers	ICR, ICRH
Interrupt enable set registers	IESR, IESRH
Interrupt enable clear registers	IECR, IECRH
Interrupt evaluate register	IEVAL

Table 9 summarizes the differences between interrupt parameters of the OPT field and newly introduced interrupt registers on EDMA3.

Table 9. Summary of Interrupt Parameters of OPT Field Differences Between EDMA3 and EDMA2

Feature	EDMA2	EDMA3
Transfer Complete Code	Two fields, two-bit TCCM and four-bit TCC field, to form the final six-bit interrupt complete code.	Single six-bit field for interrupt complete code.
Intermediate Transfer Complete Code	Uses the TCC bit for interrupt complete code and the ATCC bit for intermediate interrupt complete code.	Uses the same TCC bit for both the final and the intermediate interrupt complete code.
Pseudo Registers for Interrupt Enable Register on EDMA3	–	Introduces pseudo registers for interrupt enable registers. Instead of directly modifying the interrupt enable registers (IER and IERH), you must write to the interrupt enable set registers (IESR and IESRH) to enable any interrupt. You must also write to the interrupt enable clear registers (IECR and IECRH) to clear any bits in the interrupt enable registers.
Interrupt Clear Register (ICR and ICRH)	–	Introduces the Interrupt Clear Registers (ICR and ICRH) for clearing any bits in the Interrupt Pending Register (IPR).
Interrupt Evaluate Register (IEVAL)	–	Introduces the Interrupt Evaluate Register (IEVAL). Writing a 1 to the EVAL bit in IEVAL forces the EDMA Channel Controller (CC) to generate an interrupt if there are still interrupts pending in IPR/IPRH. You must use this feature to write robust DMA interrupt service routines.

2.7.3 Region Interrupt on EDMA3

EDMA3 introduces the concept of a shadow region. You can only access a user-defined subset of DMA and QDMA resources through each shadow region. DMA and QDMA region control registers define or assign the ownership of DMA and/or QDMA channels for each shadow region.

There are eight EDMA3 shadow regions (and its associated memory maps) on TCI648x. There is a set of registers associated with each shadow region that defines which channels and interrupt completion codes belong to that region. To assign ownership of the DMA/QDMA channels to a region, program these registers per region. For an application, a unique assignment is expected to be given to the QDMA/DMA channel (therefore, a bit position) to a specified region.

Each shadow region has its own interrupt signal on the TCI648x devices. The transfer completion of any DMA or QDMA channels belonging to a shadow region can potentially trigger the region interrupt signal for each of the shadow regions. Table 5 lists the interrupt signal for each region.

In addition to the global interrupt enable registers, the DMA and QDMA Region Access Enable registers (DRAE*n*) serve as secondary interrupt enable registers. To enable the DMA transfer completion interrupt of a particular channel of a shadow region, both the corresponding bits in the global interrupt enable register and the DMA region access enable register must be set as follows:

- EDMACC_INT0

```
(IPR.E0 & IER.E0 & DRAE0.E0) | (IPR.E1 & IER.E1 & DRAE0.E1) | …|(IPRH.E63 & IERH.E63 & DRAHE0.E63)
```

- EDMACC_INT1

```
(IPR.E0 & IER.E0 & DRAE1.E0) | (IPR.E1 & IER.E1 & DRAE1.E1) | …| (IPRH.E63 & IERH.E63 &
DRAHE1.E63)
```

- EDMACC_INT7:

```
(IPR.E0 & IER.E0 & DRAE7.E0) | (IPR.E1 & IER.E1 & DRAE7.E1) | …|(IPRH.E63 & IERH.E63 & DRAEH7.E63)
```

2.7.4 Interrupt Service Routine

The Channel Controller (CC) does not generate a new interrupt signal for any new interrupts that are pending if you have not cleared previous pending interrupts on EDMA3. The software must ensure that the Interrupt Service Routines (ISRs) service all of the pending interrupts. Comparatively, EDMA2 hardware is more forgiving. A write to clear a CIPR bit forces the EDMAINT signal to repulse if more interrupts are pending, thus minimizing the chance of losing interrupts due to race conditions.

There are two options for constructing a robust ISR for EDMA3:

- Poll all of the bits during execution of the ISR, and clear all enabled bits in the Interrupt Pending Register (IPR) by writing to the Interrupt Pending Clear register (ICR/ICRH) before exiting any ISR, as shown in the following pseudo code:
  1. Enter ISR
  2. Read IPR
  3. For the condition set in IPR,
     a. Perform operation as needed
     b. Clear bit for serviced INT
  4. Read IPR
     a. If IPR != 0, go to step 3
     b. If IPR == 0, exit ISR
- Service the ISR. Before exiting the ISR, write to the EVAL bit in the Interrupt Evaluate Register (IEVAL). Writing to the EVAL bit in the IEVAL forces the EDMA CC to generate a new interrupt signal, if there are still pending interrupts in the Interrupt Pending Register (IPR/IPRH). The following lists the pseudo code for this ISR option:
  1. Enter ISR
  2. Read IPR
  3. For the condition set in IPR,
     a. Perform operation as needed
     b. Clear bit for serviced INT
  4. Read IPR
     a. If IPR == 0, exit ISR
     b. If IPR != 0, set EVAL bit in IEVAL to force a new interrupt to trigger

2.8 Event Queue

2.8.1 Number of Event Queues and Queue Length

EDMA2 has a total of four event queues on C64x devices. Each queue length is 16.

EDMA3 has four event queues on the TCI648x devices. Each queue length is 16.

2.8.2 Available Queue Length for EDMA and QDMA

The entire queue length is allocated for DMA, QDMA, and master peripherals on EDMA2. You can allocate a maximum 7 queue length to either DMA or QDMA.

The entire queue length of 16 of all of the available event queues is allocated for DMA and QDMA channels on EDMA3. EDMA3 does not necessitate manual configuration for allocation of queue length for EDMA or QDMA.

2.8.3 Event Queue Timing

When an event queue admits a transfer request from either DMA or QDMA on EDMA3, the event queue queues the event number. The corresponding PaRAM is only free for configuration for future DMA transfer after the Transfer Controller (TC) accepts a transfer request.

The event queue stores the entire PaRAM entry table on EDMA2. The PaRAM is ready for future use as soon as the event queue admits an event.

It is important to be aware of the differences of the event queues between two DMAs. You must check to ensure that the previous QDMA transfer request has been submitted to the EDMA TC before configuring the related PaRAM for future use.

2.9 Priority Processing

2.9.1 Mapping Between DMA/QDMA Events and Event Queues

Configure the DMA Queue Number registers (DMAQNUM0-DMAQNUM1) and the QDMA Queue Number register (QDMAQNUM) on EDMA3 to assign or map 64 DMA and 4 QDMA channels to four event queues of the Channel Controller (CC).

The four event queues are categorized into urgent, high priority, medium priority, and low priority queues on EDMA2. Configure the PRI parameter in the Channel Options Parameter (OPT) of the corresponding PaRAM entry to map DMA and QDMA channels and the four event queues.

2.9.2 DMA/QDMA Queue Priorities versus Other Master Transfers

The EDMA Transfer Controller (TC) only handles DMA and QDMA transfers that you program on EDMA3. The transfer requests are eventually submitted to two transfer controllers. The transfer controllers are the masters of a Switched Central Resource (SCR). Other transfers between different masters such as EMAC, DSP, and ARM to different slaves such as DDR and asynchronous EMIF go directly through the SCR. The SCR handles the priority arbitration between DMA/QDMA transfers and other master-to-slave transfers. Use the following registers to program the priorities of each of the masters:

- Event Queue Priority registers (QUEPRI0, QUEPRI1, QUEPRI2, QUEPRI3): Program these four registers to set the priorities of the four event queues of EDMA that connect to the SCR through the EDMA transfer controller.
- Master Priority registers (MSTRPRI0, MSTRPRI1): Program these two registers to set the priorities of the masters, including Rapid I/O, PCI, HPI, and EMAC.

Contrarily, both DMA/QDMA transfers and other master-to-slave transfers go through the same event queues and TC on EDMA2. All of the transfers compete for the same bandwidth of the TC. To program the priorities, configure the PRI field of PaRAM to assign masters to different priority queues.

2.10 Linking on EDMA3

2.10.1 Linking for DMA

EDMA3 always enables linking for DMA and QDMA transfer. You must provide a valid link address in the LINK field of the corresponding PaRAM entry. Additionally, you must set the STATIC bit in OPT to 0 for all of the linked PaRAM entries, except for the last one. You must set the last linked PaRAM entry to 1 for linking on QDMA.

Linking for DMA transfer is optional on EDMA2. Set LINK = 1 in OPT to enable linking. You must provide a valid link address in the LINK field of the PaRAM entry. Linking is not available for QDMA on EDMA2. The link field of the QDMA registers is reserved.

2.10.2 Terminating Linking

Terminate linking by linking a parameter entry to NULL on both EDMA2 and EDMA3.

It is not necessary to define a NULL entry explicitly on EDMA3. It is only necessary to set the LINK field of the last valid PaRAM entry to FFFFh. The Channel Controller (CC) automatically updates all of the fields of the mapped PaRAM entry to zeros, except for the LINK field with a value of FFFFh. You must also set the STATIC bit in the Channel Options Parameter (OPT) of the last valid PaRAM entry to 1 on QDMA.

A NULL entry is a PaRAM entry with all of the fields cleared to 0 on EDMA2. You must define and create a NULL entry. You must link its last valid PaRAM entry to this NULL entry to terminate linking.

Table 10. Comparison of Parameters for Chaining for EDMA2 and EDMA3

Feature	Description	EDMA2	EDMA3
Chaining/Interrupt	Chain Enabling bit in OPT	TCINT	TCCHEN
	Interrupt Enabling bit in OPT	TCINT	TCINTEN
	Transfer Complete code	TCCM:TCC	TCC
Intermediate Chaining/Interrupt	Intermediate Chain Enabling bit in OPT	ATCINT	ITCCHEN
	Intermediate Interrupt Enabling bit in OPT	ATCINT	ITCINTEN
	Intermediate Transfer Complete code	ATCC	TCC

2.11 Chaining on EDMA3

EDMA2 and EDMA3 have similar chaining functionalities. When the DMA transfer completes, the chain event register of a DMA channel is set if chaining is enabled. The chain event register of a DMA channel is set as specified by the TCC value in the Channel Options Parameter (OPT) and triggers another transfer request. Although the chaining functionality is the same for both EDMA2 and EDMA3, there are some differences, as summarized in Table 10.

The Transfer and Intermediate Transfer Complete Chaining Enabling (TCCHEN and ITCCHEN) bits are different from the bits for Transfer and Intermediate Transfer Complete Interrupt (TCINTEN and ITCINTEN) on EDMA3. The TCCHEN and ITCCHEN bits in OPT enable transfer and intermediate transfer complete chaining. The TCINTEN and ITCINTEN bits in OPT enable TCINTEN and ITCINTEN.

The Transfer and Intermediate Transfer Complete Chaining Enabling (TCINT and ATCINT) bits are the same as those for Transfer and Intermediate Transfer Complete Interrupt Enabling bits on EDMA2. Additionally, there are separate registers for enabling interrupt and for enabling chaining on EDMA2. The registers are: Channel Interrupt Enable Register (CIER) for enabling interrupt and Chain Enable Register (CCER) for enabling chaining. Having separate bits for enabling chaining and for enabling interrupt offers more flexibility for software development on EDMA3. Two different DMA channels can use the same TCC code for different purposes. On one DMA channel, enabling chaining and disabling transfer completion interrupt can occur concurrently. Further, enabling completion interrupt and disabling transfer chaining can occur simultaneously. This feature provides more flexibility and freedom for software development. Using the same TCC value for two different channels to achieve the same functionality is not possible on EDMA2. Both enabling chaining and interrupt use the same bits.

2.12 Set and Clear Operation on EDMA3

EDMA3 introduces pseudo registers for reliable operation in multitasking and multiprocessor environments. Instead of directly modifying interrupt and event registers, these registers are set or cleared through their pseudo registers. This new feature eliminates race conditions when multiple processors try to access the same DMA register at the same time. The following list summarizes the pseudo registers that EDMA3 introduces.

- EESR/EESRH and EECR/EECRH for the Event Enable Register (EER/EERH): Set the Event Enable Register (EER*n*/EERH*n*) through the Event Enable Set Register (EESR/EESRH). Clear the EER*n* through the Event Enable Clear register (EECR/EECRH).
- QEESR/QEESRH and QEECR/QEECRH for QDMA Event Enable Register (IER/IERH): Set the IER/IERH registers through the QDMA Set Registers (QEESR/QEESRH). Clear the IER/IERH registers via the QDMA Clear Registers (QEECR/QEECRH).
- IESR and IECR for Interrupt Enable Register IER/IERH. Enable the Interrupt Enable Register (IER/IERH) through the Interrupt Set Enable Register (IESR). Clear the IER/IERH through the Interrupt Enable Clear Register (IECR)

3 New Features on EDMA3

3.1 DMA Channel Region Access

EDMA3 introduces the concept of DMA channel region access. Three types of regions are defined for the Channel Controller (CC) on EDMA3:

1. Global regions
2. Global DMA channel region
3. DMA channel shadow regions

The global region and the global DMA channel region consists of all of the memory-mapped registers for all of the DMA channels, QDMA channels, and interrupt.

The DMA channel shadow regions provide restricted view or access to the DMA and/or QDMA registers in the global DMA channel region. There are eight shadow regions on EDMA3. Each region has a Region Access Enable register (DRAE*n*/DRAEH*n*) and QDMA Region Access Enable register (QRAE*n*). The region access enable registers define the ownership or the right of access of the shadow regions to the DMA and QDMA channel registers in the global DMA channel region.

A value of 1 in a bit position on the DMA/QDMA region access enable register implies that the ownership of that channel and that write accesses to the corresponding bit position are allowed to take effect. It also implies that reads should return valid data. A value of 0 means that writes to that bit position are discarded without modifying the original register value and reads return a value of 0.

3.2 Memory Protection

Memory protection on EDMA3 allows the EDMA Channel Controller (CC) to offer restricted access to its internal resources. Memory protection also allows the CC to accept or deny access to its internal resources, based on the following permission characteristics (you program these permission characteristics):

1. Requestor ID
2. Privilege level: User or Supervisor
3. Access Type: Read (R), Write (W), and Execute (X)

The CC divides its resources into multiple regions, including a global region and up to eight shadow regions.

Each region has its associated Memory Protection Page Attribute (MPPA) register. The MPPA registers define the memory protection characteristics for their corresponding regions. For more information on memory protection, see the *TMS320TCI648x DSP Enhanced Direct Memory Access (EDMA3) Controller User's Guide* (SPRU727).

3.3 Debugging Capabilities

3.3.1 Debug Visibility

3.3.1.1 Event Queue Visibility

EDMA3 allows monitoring of the history of 16 events in every event queue. The EDMA Channel Controller (CC) records the events in each event queue in the Event Queue Entry registers (Q*k*E*n*), where *k* = 0-1 and *n* = 0-15, *k* identifies the event queue, and *n* indicates the order of the event entry.

The Q*k*E*n* registers capture the type of event. Types of events that the Q*k*E*n* registers capture include events triggered via ER, manual events triggered via ESR, chain events triggered via CER, and auto triggered events via the QER, and the event number.

3.3.1.2 Event Queue Watermarking

The event queue watermarking feature on EDMA3 monitors the maximum number of event entries in each of the event queues. The watermarking information is captured in the WM field of the Event Queue Status register (QSTAT*n*).

3.3.1.3 Event Queue Threshold

The EDMA3 Channel Controller (CC) keeps track of whether the number of event entries in an event queue exceeds the threshold. You program the queue threshold for each event queue in the queue threshold *x* registers (QWMTHRA and QWMTHRB). A queue threshold error occurs when the number of event entries in any of the event queues exceeds the threshold that you programmed. The error is captured in the THRXCD field of the Event Queue Status Register (QSTAT*n*) and in the QTHRXCD*n* field in the EDMA3CC Error Register (CCERR).

3.3.2 Error Detection

3.3.2.1 Event Missed Registers

EDMA3 introduces the Event Miss Register (EMR/EMRH) and the QDMA Event Miss Register (QEMR).

For EMR/EMRH, if any DMA channel event happens before the channel's previous event has been cleared or submitted to the Transfer Controller (TC), the event is considered a missed event and is captured in the corresponding bit of EMR or EMRH. You can clear the bits in EMR/EMRH by writing a 1 to the corresponding bit in the Event Miss Clear Register (EMCR/EMCRH).

For QEMR, if any QDMA channel event happens before the channel's previous event has been cleared or submitted to the TC, the event is considered a missed event and is captured in the corresponding bit of QEMR. You can clear the bits in EMR/EMRH by writing a 1 to the corresponding bit in the QDMA Event Miss Clear Register (QEMCR).

3.3.2.2 Channel Controller Error Registers

The EDMA3CC Error Register (CCERR) captures the queue threshold error and the transfer completion code error.

3.3.2.3 Memory Protection Error Registers

Whenever there is a memory protection violation on EDMA3, the MPFAR register captures the faulting address and the MPFSR register captures the access violation type, including Read, Write, Execute, and privilege level violation either by User or by Supervisor.

3.3.2.4 Error from Null Parameter Entry

If an event is triggered for a NULL PaRAM entry on EDMA3, an error occurs. When this error occurs, the EDMA Channel Controller (CC) sets the corresponding bits in both the Secondary Event Register (SER) and the Event Miss Register (EMR).

When a 1 is set in the SER, the CC disables the corresponding DMA or EDMA channel for future use. You must clear both bits in SER and EMR to bring the disabled DMA or QDMA channel back.

3.3.2.5 Error Interrupts

Whenever there is a DMA missed event error, QDMA missed event error, and queue threshold error, the EDMA3CC generates an interrupt signal via EDMA3CC_ERRINT.

Whenever there is a memory protection violation, the EDMA3CC issues an interrupt signal via EDMA3CC_ MPINT.

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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