

TMS320TCI6488 Memory Access Performance

Communication Infrastructure

Brighton Feng

Abstract

The TMS320TCI6488 has three C64x+ cores, each of which has 32KB L1D SRAM, 32KB L1P SRAM, 3MB L2 SRAM and can be configured as 1MB/1 MB/1 MB or 1.5 MB/1 MB/0.5 MB among the three DSP cores. A 32-bit 667MHz DDR2 SDRAM interface is provided on the DSP to support up to 512MB of external memory.

Memory access performance is very critical for software running on the DSP. On the TCI6488 DSP, all the memories can be accessed by DSP cores and multiple DMA masters.

Each DSP core is capable of performing up to 128 bits of load/store operations per cycle. When accessing L1D SRAM the DSP core can access the memory at up to 16GB/second at a 1GHz core clock frequency.

The DMA switch fabric, which provides the interconnection between the C64x+ cores (and their local memories), external memory, the EDMA controllers, and on-chip peripherals, has access port to each end point, which is capable of sustaining up to 5.333GB/second at a 1GHz core clock frequency. There are six EDMA transfer controllers that can be programmed to move data, concurrently, between any memory endpoints on the device.

This document gives designers a basis for estimating memory access performance and provides measured performance data achieved under various operating conditions. Some factors affecting memory access performance are discussed.

Contents

1	Introduction	3
2	DSP Core vs. EDMA3 and IDMA for Memory Copy	6
3	Performance of DSP Core Access Memory	8
3.1	Performance of DSP Core Access L2	8
3.2	Performance of DSP Core Access External DDR2 Memory	9
4	Performance of DMA Access Memory	14
4.1	DMA Transfer overhead	14
4.2	EDMA Performance Difference Between 6 Transfer Engines	14
4.3	EDMA Bandwidth vs. Transfer Flexibility	15
4.3.1	First Dimension Size (ACNT) Considerations, Burst Width	15
4.3.2	Two Dimension Considerations, Transfer Optimization	16
4.3.3	Index Consideration	17

4.3.4 Address Alignment	18
5 Performance of Multiple Masters Sharing DDR2	19
5.1 Performance of Multiple DSP Cores Sharing DDR2	21
5.2 Performance of Multiple EDMA Sharing DDR2	21
5.3 Effect of DDR2 Bank and Row Switch on Multiple Masters Sharing DDR2	23
5.4 Effect of Priority on Multiple Masters Sharing DDR2	24
6 References	26

Figures

Figure 1 TMS320TCI6488 Memory System	4
Figure 2 DSP Core Access L2	8
Figure 3 DSP Core Reads from DDR2	10
Figure 4 DSP Core Writes to DDR2	12
Figure 5 DSP Core Read vs. Writes to Non-Cacheable DDR2	13
Figure 6 Effect of ACNT Size on EDMA Bandwidth	16
Figure 7 Linear 2D Transfer	17
Figure 8 Index Effect on EDMA Transfer	18
Figure 9 DDR2 architecture and access paths	19

Tables

Table 1 Theoretical Bus Bandwidth of DSP Core, IDMA and EDMA	6
Table 2 Maximum Throughput of Different Memory Endpoints	6
Table 3 Transfer Bandwidth Comparison Between DSP Core, EDMA and IDMA	6
Table 4 EDMA Transfer Overhead	14
Table 5 IDMA Transfer Overhead	14
Table 6 Difference Between TCs	14
Table 7 Throughput comparison between TCs	15
Table 8 Data Organization on DDR2 Banks	19
Table 9 Multiple Master Access Different Rows on Same DDR2 Bank	20
Table 10 Multiple Master Access Different Rows on Different DDR2 Banks	20
Table 11 Performance of Multiple DSP Cores Sharing DDR2	21
Table 12 Performance of Multiple EDMA Sharing DDR2	22
Table 13 Probability of Multiple Masters Access Same DDR2 Bank	23
Table 14 Organize Data Buffer in Different Banks to Alleviate Row Switch Overhead	24
Table 15 Effect of Priority on Multiple EDMA TCs Access to DDR2	24
Table 16 Effect of BPRI0 on Multiple EDMA TCs Access to DDR2	25

1 Introduction

The TMS320TCI6488 has three C64x+ cores. Each of them has:

- 32KB L1D (Level 1 Data) SRAM, which runs at the DSP Core speed, and can be used as normal data memory or cache.
- 32KB L1P (Level 1 Program) SRAM, which runs at the DSP Core speed, and can be used as normal program memory or cache.

The TCI6488 contains 3MB L2 (Level 2) unified SRAM. The L2 SRAM can be configured as 1 MB/1 MB/1 MB or 1.5 MB/1 MB/0.5 MB among the three DSP cores at boot time. The L2 SRAM runs at the DSP Core speed divided by two, and can be used as normal memory or cache for data and program.

A 32-bit 667MHz DDR2 SDRAM interface is provided on the DSP to support up to 512MB external memory, which can be used as data or program memory.

Memory access performance is very critical for software running on the DSP. On the TCI6488 DSP all the memories can be accessed by the DSP cores and multiple DMA masters.

Each TMS320C64x+ core has the ability to sustain up to 128 bits of load/store operations per cycle to the level-one data memory (L1D), and is capable of handling up to 16GB/second. When accessing data in the level-two (L2) unified memory or external memory, the access rate depends on the memory access pattern and cache.

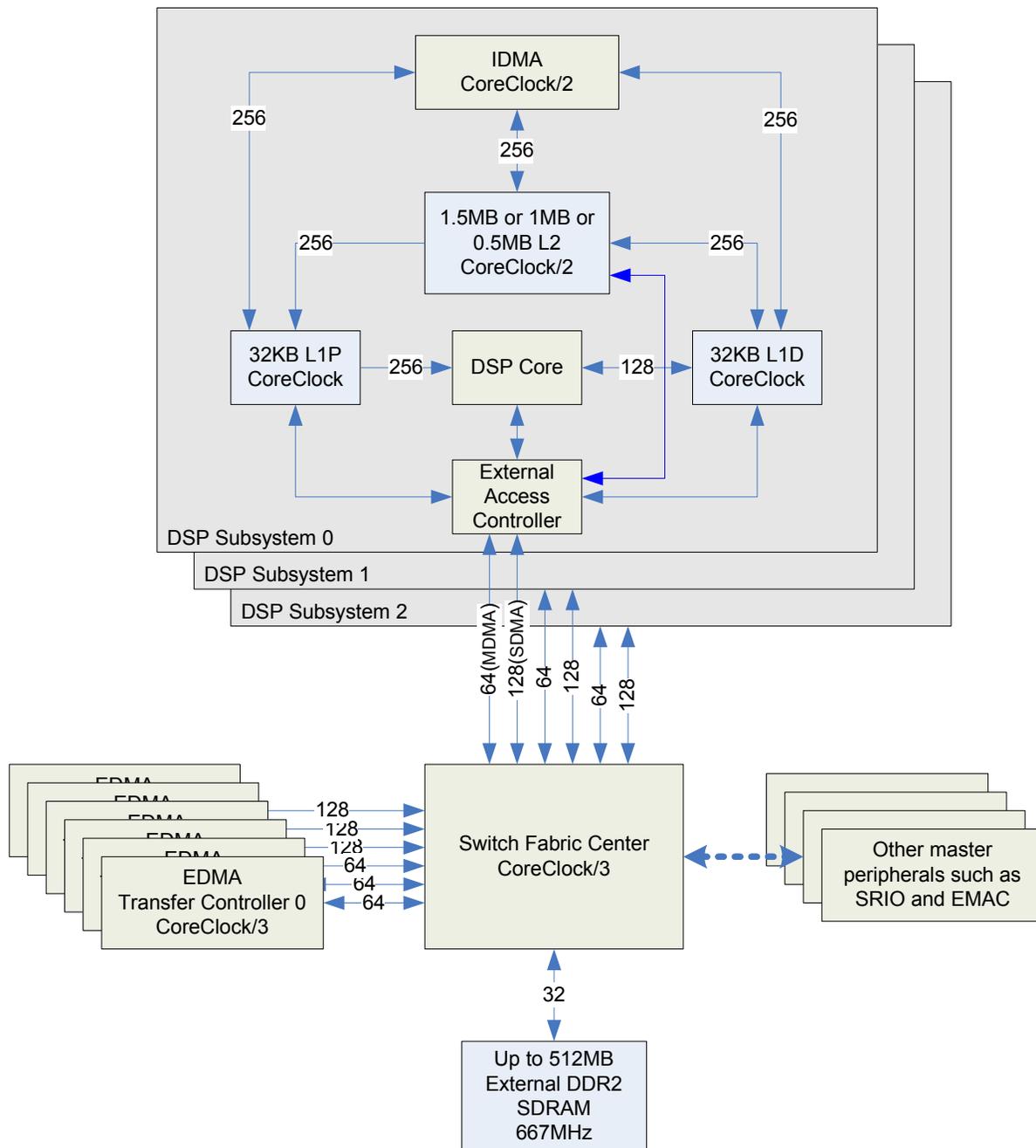
There is an internal DMA (IDMA) engine that can move data at a rate of the DSP Core speed divided by two, capable of handling up to 16GB/s, in the background of DSP core activity (i.e. data can be brought in to buffer A while the DSP core is accessing buffer B). The IDMA can only transfer data between level-one (L1), level-two (L2) and a peripheral configuration port, it cannot access external memory.

The DMA switch fabric, which provides the interconnection between the C64x+ cores (and their local memories), external memory, the enhanced DMA v3 (EDMA3) controllers, and on-chip peripherals, has a 64-bit or 128-bit access bus to each end point, runs at the DSP core frequency divided by three. Therefore, in theory, it is capable of sustaining up to 2.666GB or 5.333GB/second at a 1GHz core clock frequency.

There are six EDMA TC (Transfer Controllers) that can be programmed to move data, concurrently, in the background of DSP core activity, between the on-chip level-one (L1) memory, level-two (L2) memory, external memory, and the peripherals on the device. Each has a 64-bit or 128-bit data bus, and each transfer engine is capable of handling up to about 2.666GB or 5.333GB/second of data throughput at a core rate of 1GHz. The EDMA3 architecture has many features designed to facilitate simultaneous multiple high-speed data transfers. With a working knowledge of this architecture and the way in which data transfers interact and are performed, it is possible to create an efficient system and maximize the bandwidth utilization of the EDMA3.

[Figure 1](#) shows the memory system of the TMS320TCI6488. The number on the line is the bus width. Most modules run at CoreClock/n. The DDR2 can run up to 667MHz.

Figure 1 TMS320TCI6488 Memory System



This document gives designers a basis for estimating memory access performance, and provides measured performance data achieved under various operating conditions. Most of the tests operate under best-case situations to estimate maximum throughput that can be obtained. The transfers described in this document serve as a sample of interesting or typical performance conditions.

Some factors affecting memory access performance are discussed in this document, such as access stride, index, and conflict, etc.

This document should be helpful for analyzing the following common questions:

1. Should I use the DSP core or DMA for data copy?
2. How many cycles will be consumed for my function with many memory accesses?
3. How much degradation will be caused by multiple masters sharing the memory?

Most of the performance data in this document is examined on the 1GHz TCI6488 EVM with 32-bit 667MHz DDR2 memory.

2 DSP Core vs. EDMA3 and IDMA for Memory Copy

The bandwidth of memory copy is limited by the worst of the following three factors:

1. Bus bandwidth
2. source throughput
3. destination throughput

Table 1 summarizes the theoretical bandwidth of the C64x+ core, IDMA, and EDMA on a 1GHz TCI6488.

Table 1 Theoretical Bus Bandwidth of DSP Core, IDMA and EDMA

Master	Maximum bandwidth MB/s	Comments
C64x+ core	16000	(128 bits)/(8 bit/byte)*1000M= 16000MB/s
IDMA	16000	(256 bits)/(8 bit/byte)*(1000M/2) = 16000MB/s
EDMA TC0~2	2666	(64 bits)/(8 bit/byte)*(1000M/3)= 2666MB/s
EDMA TC3~5	5333	(128 bits)/(8 bit/byte)*(1000M/3)= 5333MB/s

Table 2 summarizes the theoretical throughput of different memories on a 1GHz TCI6488 EVM with 32-bit 667MHz DDR2 external memory.

Table 2 Maximum Throughput of Different Memory Endpoints

Master	Maximum bandwidth MB/s	Comments
L1D	32000	(256 bits)/(8 bit/byte)*(1000M) = 32000MB/s
L1P	32000	(256 bits)/(8 bit/byte)*(1000M) = 32000MB/s
L2	16000	(256 bits)/(8 bit/byte)*(1000M/2) = 16000MB/s
DDR2	2667	(32 bits)/(8 bit/byte)*(667M)=2667MB/s

Table 3 shows the transfer bandwidth measured for linear memory block copy with EDMA, IDMA, and DSP Core for different scenarios. The bandwidth is measured by taking the total bytes transferred and dividing it by the time it used.

Table 3 Transfer Bandwidth Comparison Between DSP Core, EDMA and IDMA

Bandwidth(MB/s) for Src-> Dst	DSP core	EDMA	IDMA
L1D-> L1D (16KB L1D cache)	7606	3347	3762
L2-> L1D (16KB L1D cache)	3161	3345	7037
L1D-> L2 (16KB L1D cache)	7798	3575	9092
L2-> L1P (16KB L1P cache)	N/A	3369	7037
L2 -> L2 (32KB L1D cache)	2459	3547	3409
L2-> L2 of another core (non-cacheable)	444	3547	N/A
L2-> L2 of another core (32KB L1D cache)	444		N/A
L2-> L2 of another core (32KB L1D, 256KB L2 cache)	496		N/A
L2 of another core-> L2 (non-cacheable)	93		N/A
L2 of another core-> L2 (32KB L1D cache)	561		N/A
L2 of another core-> L2 (32KB L1D, 256KB L2 cache)	761		N/A

Table 3 Transfer Bandwidth Comparison Between DSP Core, EDMA and

Bandwidth(MB/s) for Src-> Dst	DSP core	EDMA	IDMA
L2 -> DDR2 (non-cacheable)	435	2580	N/A
L2 -> DDR2 (32KB L1D cache)	435		N/A
L2 -> DDR2 (32KB L1D, 256KB L2 cache)	503		N/A
DDR2 -> L2 (non-cacheable)	98	2060	N/A
DDR2 -> L2 (32KB L1D cache)	561		N/A
DDR2 -> L2 (32KB L1D, 256KB L2 cache)	771		N/A
DDR2 -> DDR2 (non-cacheable, src/dst in same bank)	70	952	N/A
DDR2 -> DDR2 (32KB L1D cache, src/dst in same bank)	203		N/A
DDR2 -> DDR2 (32KB L1D, 256KB L2 cache, same bank)	257		N/A
DDR2 -> DDR2 (non-cacheable, src/dst in different bank)	76	976	N/A
DDR2 -> DDR2 (32KB L1D cache, src/dst in different bank)	232		N/A
DDR2 -> DDR2 (32KB L1D, 256KB L2 cache, diff bank)	268		N/A

Generally speaking, the core accesses internal memory efficiently, while using the DSP core to access external data is a bad use of resources and should be avoided. The IDMA is good at linearly moving a block of data in internal memory (L1D, L1P, L2), but it cannot access external memory. The EDMA3 should be given the task of transferring data to/from external memory.

The cache configurations dramatically affect the DSP core performance, but it does not affect EDMA and IDMA performance. All test data for the DSP core in this application note are based on cold cache, i.e., all the caches are flushed before the test.

The DDR2 bank architecture affects the performance slightly for the DDR2->DDR2 case. If the source and destination are in the same bank, the DDR2 row switch happens frequently, every row switch introduces extra delay cycles; if the source and destination are in a different bank, the frequency of row switch is reduced, thus improving the performance.

The above EDMA throughput data is measured on TC3 (Transfer Controller 3). The TC4 and TC5 have the same performance, while the throughput of TC0, TC1, and TC2 is not as good as TC3. See more details in the following section for the comparison between four DMA transfer engines.

3 Performance of DSP Core Access Memory

L1 runs at the same speed as the DSP core, so the DSP core can access L1 memory one time per cycle. For special applications that require accessing a small data block very quickly, part of the L1 can be used as normal RAM to store the small data block.

Normally, L1 is used as cached. If a cache hit happens, the DSP core can access data in one cycle. If a cache miss happens, the DSP core stalls until the data comes into the cache.

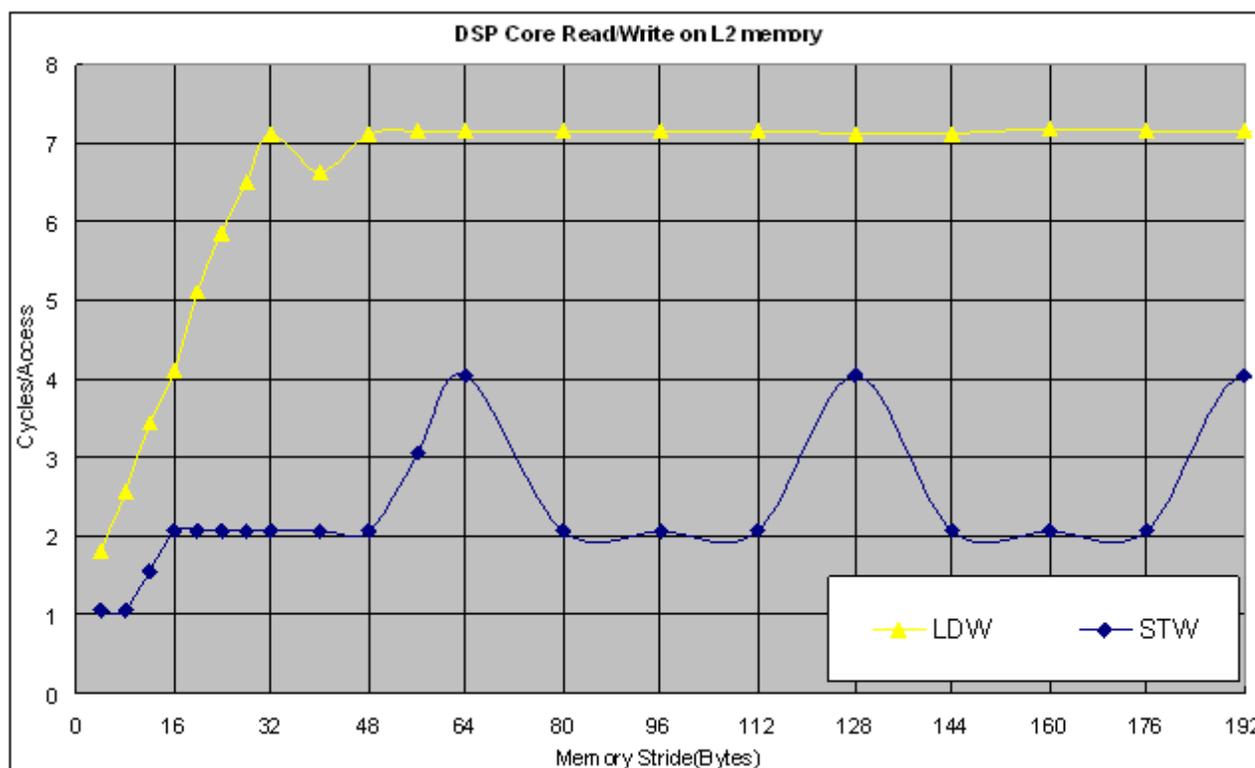
The following sections examine the access performance for DSP Core accesses of L2 and external DDR2 memory. The pseudo codes for this test appears as follows:

```
flushCache();
preCycle= getTimeStampCount();
for(i=0; i< accessTimes; i++)
{
    Access Memory at address;
    address+= stride;
}
cycles = getTimeStampCount()-preCycle;
cycles/Access= cycles/accessTimes;
```

3.1 Performance of DSP Core Access L2

Figure 2 shows data collected from a 1GHz TCI6488 EVM. The time required for 1024 consecutive LDW (LoaD Word) or STW (STore Word) instructions was measured, and the average time for each instruction is reported. The cycles for LDB/STB, and LDDW/STDW are the same as for LDW/STW.

Figure 2 DSP Core Access L2



Since the L1D is a read-allocate cache, a DSP core read of L2 should always go through L1D cache. So, a DSP core access L2 highly depends on the cache. The address increment (or memory stride) affects cache utilization. Contiguous accesses utilize cache to the fullest. A memory stride of 64 bytes or more causes every access to miss in the L1 cache because the L1D cache line size is 64 bytes.

Since the L1D is not a write-allocate cache, and the cache is flushed before the test, any write to the L2 goes through the L1D write buffer (4x16bytes). For a write operation, if stride is less than 16 bytes, several writes may be merged into one write to the L2 in the L1D write buffer, thus achieving an efficiency close to 1 cycle/write. When the stride is a multiple of 64 bytes, every write accesses the same bank of L2 (because the L2 is organized as 4 x 16 bytes banks), which requires 4 cycles. For other strides, the consecutive writes access to different banks of L2, they can be overlapped with pipeline, which requires 2 cycles only.

3.2 Performance of DSP Core Access External DDR2 Memory

DSP core access of external DDR2 memory highly depends on the cache. When the DSP core accesses external memory spaces, a TR (transfer request) may be generated (depending on whether the data are cached) to the Switch Fabric Center. The TR will be for one of the following:

- a single element - if the memory space is non-cacheable
- an L1 cache line - if the memory space is cacheable and the L2 cache is disabled
- an L2 cache line - if the memory space is cacheable and L2 cache is enabled

No transfer request is generated in the case of an L1 or L2 cache hit.

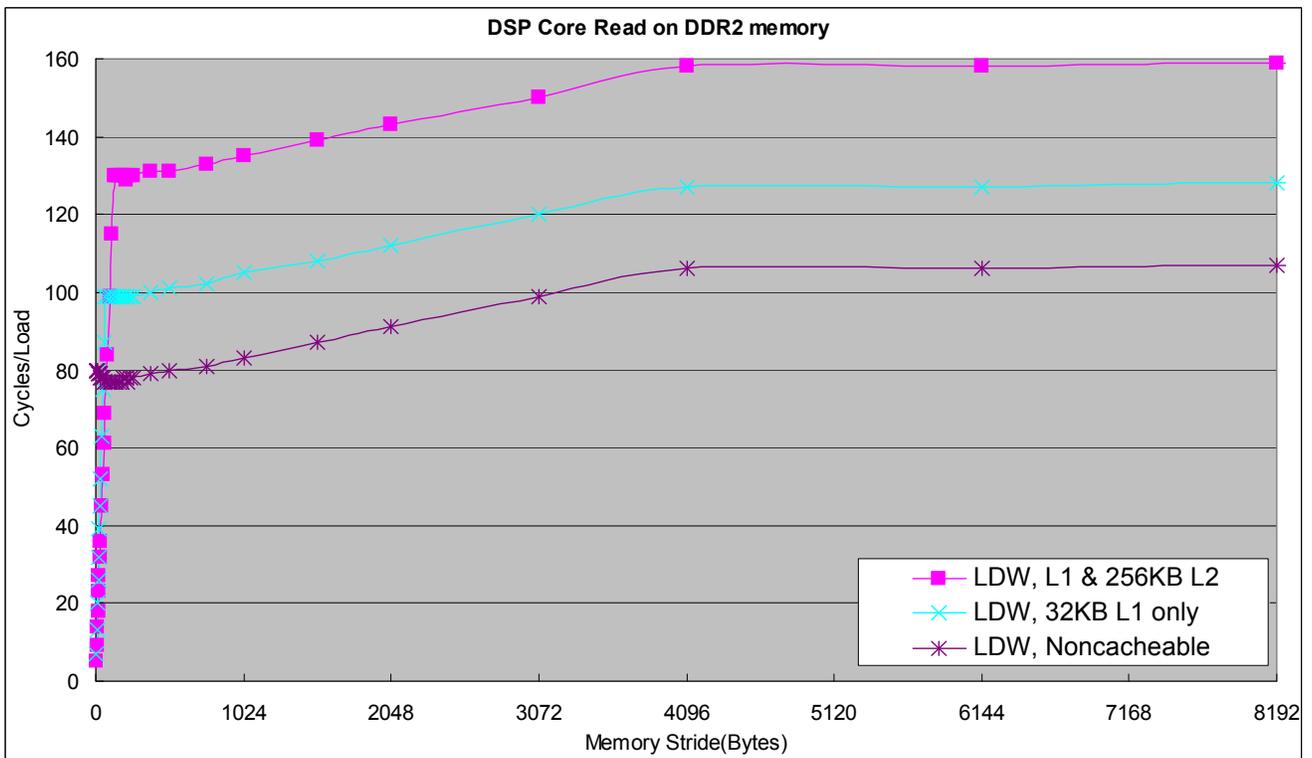
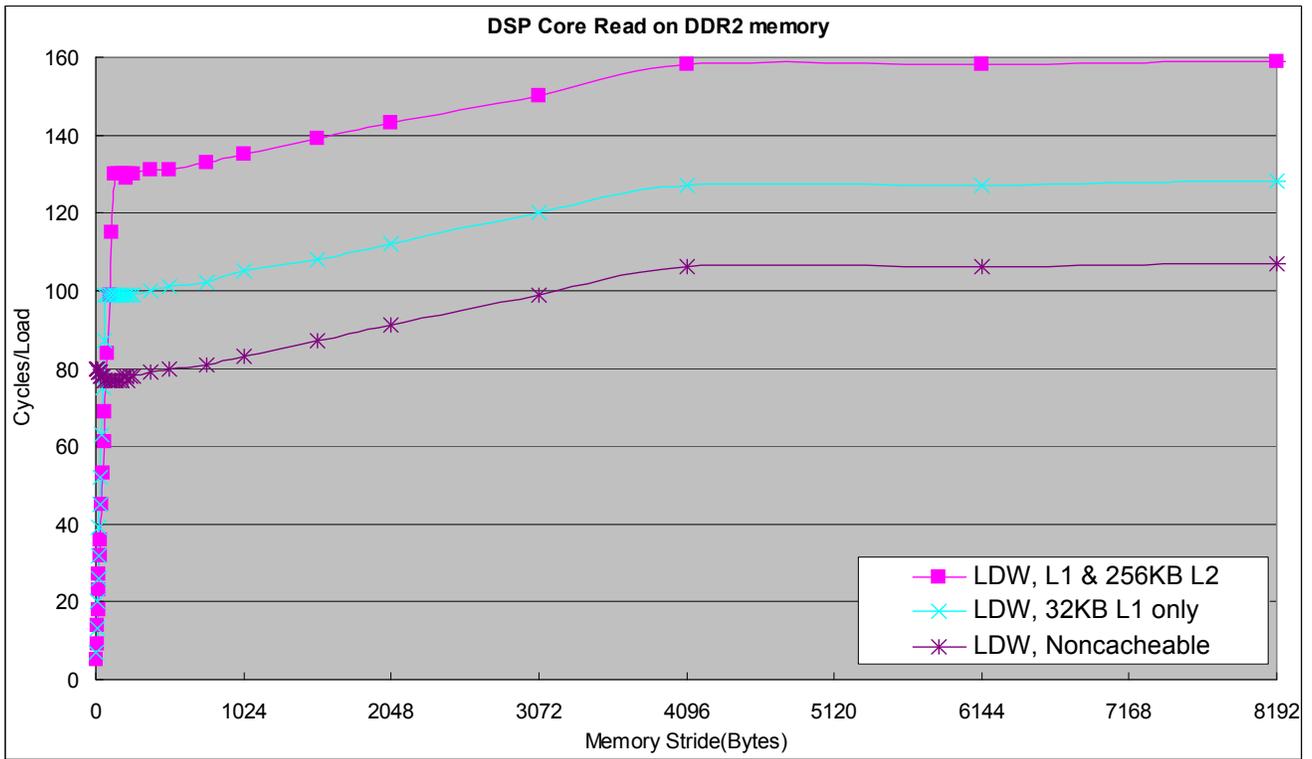
An external memory can be cached by L1 cache, L2 cache, or neither. If the appropriate MAR bit for a memory space is not set, it is not cacheable. If the MAR bit is set and L2 cache size is zero (all L2 is defined as SRAM), the external memory space is cached by L1. If the MAR bit is set and L2 cache size is greater than 0, the external memory space is cached by L2 or L1.

The address increment (or memory stride) affects cache utilization. Contiguous accesses utilize cache memory to the fullest. A memory stride of 64 bytes or more causes every access to miss in the L1 cache because the L1 line size is 64 bytes. A memory stride of 128 bytes causes every access to miss in L2 because the L2 line size is 128 bytes.

If cache miss happens, the DSP Core will stall, waiting for the return data.

[Figure 3](#) shows data collected from 1GHz TCI6488 EVM with 32-bit 667MHz DDR2. The time required for 1024 LDW (Load Word) instructions was measured, and the average time for each instruction is reported. The cycles for LDB, and LDDW are the same as LDW.

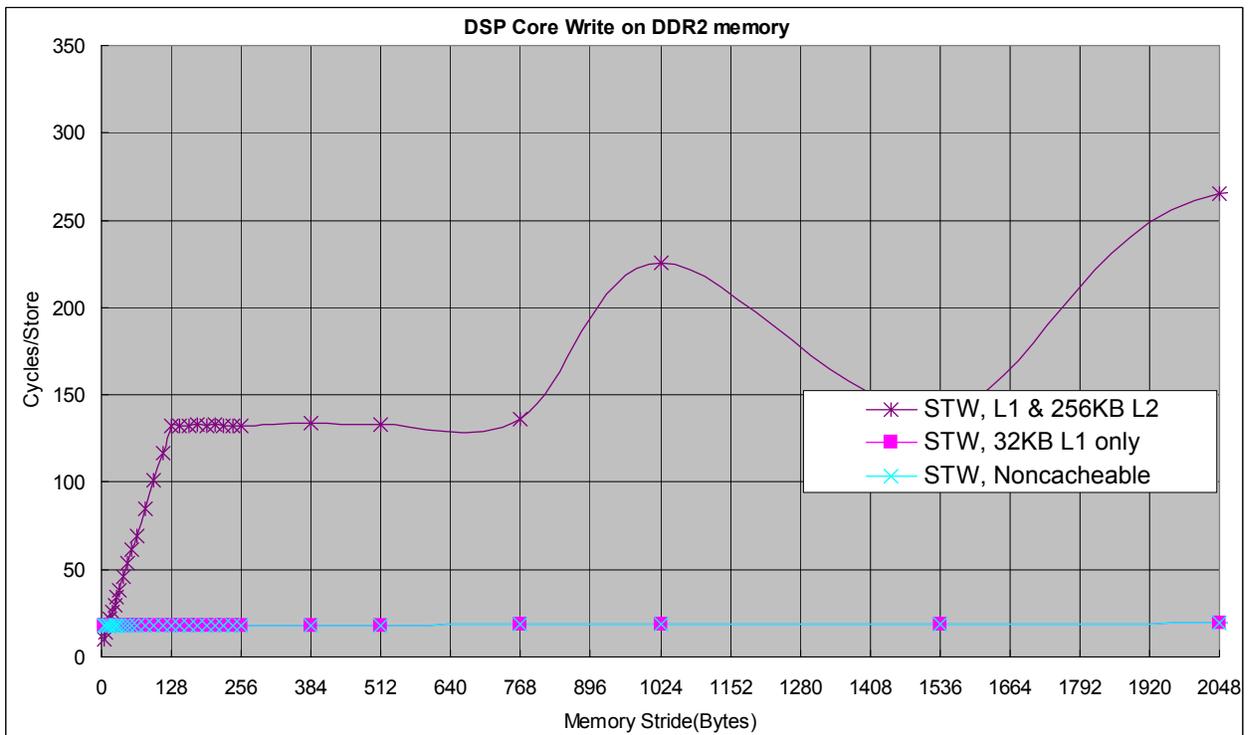
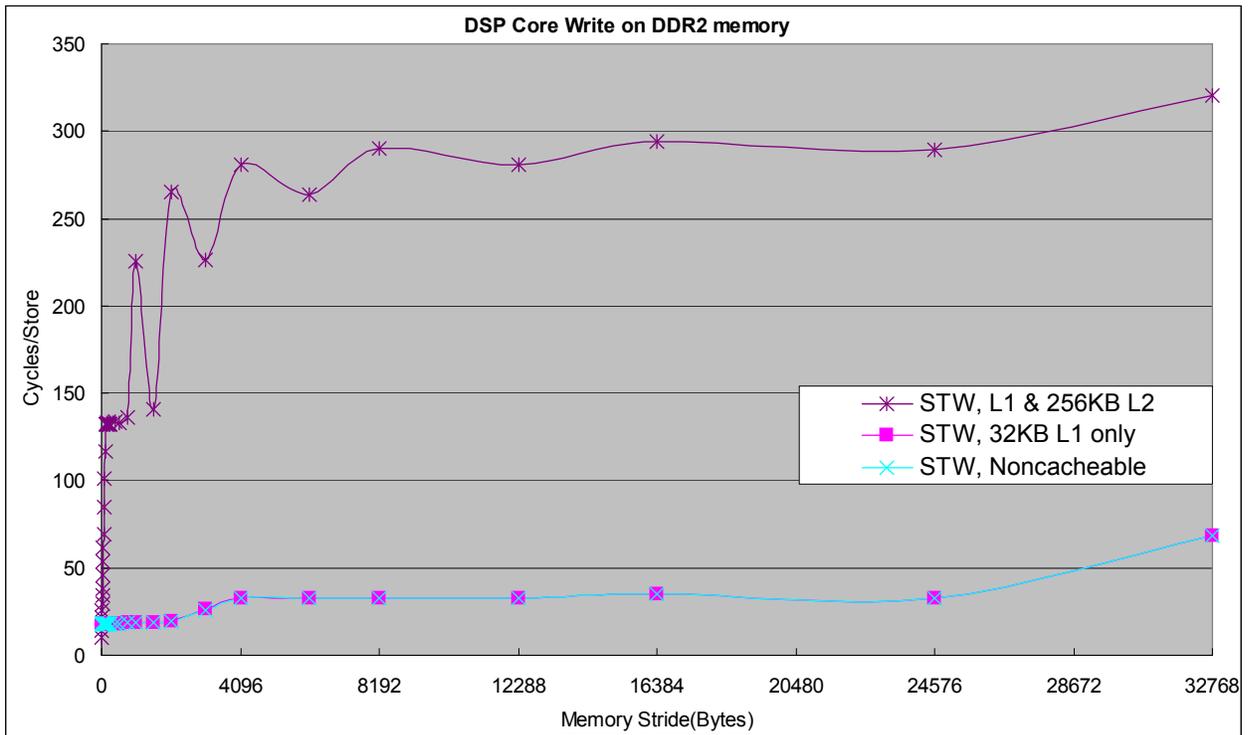
Figure 3 DSP Core Reads from DDR2



For a memory stride of less than 128 bytes, the performance is dominated by cache as discussed above. For memory stride larger than 128, the performance becomes worse because of DDR2 SDRAM row switch. The row size or bank width on the TCI6488 EVM is 4096 bytes, so for a stride larger than 4096, and every read access to a new row, the row switch results in about 30 extra cycles. Please note that the DDR2 SDRAM row switch overhead may be different for different DDR2 SDRAM.

[Figure 4](#) shows data collected from a 1GHz C64x+ writing to 32-bit, 667MHz external DDR2 memories. The time required for 1024 STW (STore Word) instructions was measured, and the average time for each instruction is reported. The cycles for STB and STDW are the same as for STW.

Figure 4 DSP Core Writes to DDR2



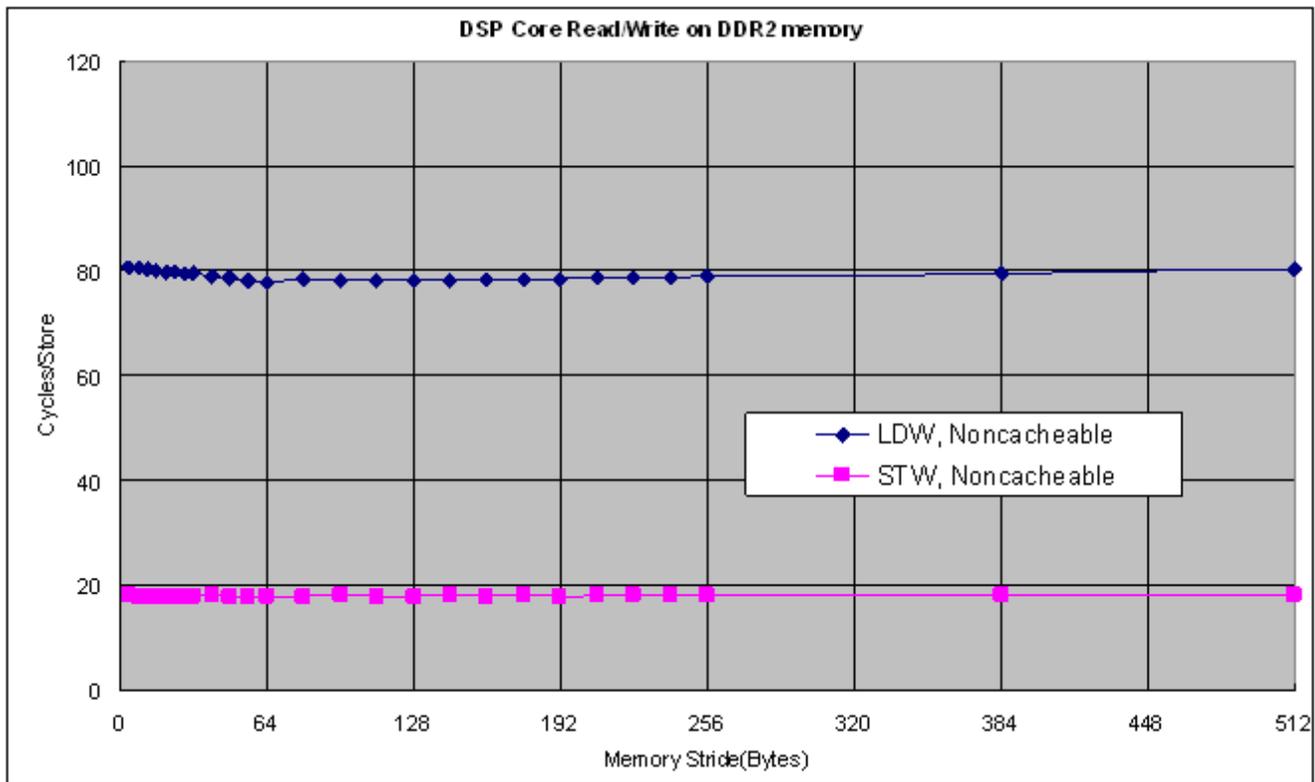
Since the L1D cache is not a write-allocate cache, and the cache is flushed before this test, it has no effect on the write operation. Therefore, the “32KB L1 only” case and the “Noncacheable” case are overlapped in the above figure.

Similar to the cacheable read operation, for the cacheable write operation with a memory stride of less than 128 bytes, the performance is dominated by cache as discussed above. For a memory stride larger than 4096 bytes, the performance becomes worse because of DDR2 SDRAM row switch.

L2 cache is a write-allocate cache. For any write operation, it always reads the 128 bytes including the accessed data into a cache line first, and then modifies the data in the L2 cache. This data will be written back to real external memory if a cache conflict happens or by manual writeback. When the memory stride is equal to or larger than 1024 bytes, the cycles needed for a write operation increases to about 2 times of the read operation, because the conflict happens frequently for a big memory stride. Thus every write operation results in a cache line write back (for conflict) and a cache line read (for write-allocate).

For non-cacheable access, cycles for a write are about 1/4 of the cycles for a read as compared in Figure 5, because four write operations are combined into one transfer request in the L1D write buffer, while four read operations submit four transfer requests to the switch fabric center.

Figure 5 DSP Core Read vs. Writes to Non-Cacheable DDR2



4 Performance of DMA Access Memory

The EDMA3 architecture has many features designed to facilitate simultaneous multiple high-speed data transfers. Its performance affected by the memory type and many other factors discussed in the following sections.

4.1 DMA Transfer overhead

Initial latency is defined as the time between when a DMA event occurs and when real data transfer begins. Since initial latency is hard to measure, we measured transfer overhead instead. This value is defined as the sum of the latency and the time to transfer the smallest element. The values vary based on the type of source/destination peripheral and readiness of the source/destination ports and peripherals. The following tables show the average cycles measured on a 1GHz TCI6488 EVM for the smallest transfer (1 word) between different ports.

Table 4 EDMA Transfer Overhead

	Destination	L1D	L2	DDR2
Source				
L1D		195	195	212
L2		195	195	221
DDR2		221	221	271

Table 5 IDMA Transfer Overhead

	Destination	L1D	L2
Source			
L1D		96	95
L2		95	99

In conclusion, transfer overhead is a big concern for short transfers and needs to be included when scheduling DMA traffic in a system. Single-element transfer performance will be latency-dominated. So, for small transfers, you should make the trade-off between DMA and DSP core.

4.2 EDMA Performance Difference Between 6 Transfer Engines

EDMA3 on a TCI6488 includes six TCs (transfer controllers). If six TCs run simultaneously, the overall throughput for EDMA3 is $3 \times (128/8 \times 1000/3) + 3 \times (64/8 \times 1000/3) = 24000 \text{ MB/s}$.

The six transfer engines are not exactly same. [Table 6](#) is a summary of the difference.

Table 6 Difference Between TCs

Name	TC0	TC1	TC2	TC3	TC4	TC5
FIFO Size	512 bytes					
Bus Width	64 bits	64 bits	64 bits	128 bits	128 bits	128 bits
Destination FIFO Entries	4 entries					
Default Burst Size	64 bytes					

For more information about the difference, please refer to the TMS320TCI6488 DSP Enhanced DMA (EDMA3) Controller User's Guide (SPRUEE9).

Table 7 compares the maximum throughput measured for different TCs.

Table 7 Throughput comparison between TCs

Bandwidth(MB/s)	TC0	TC1	TC2	TC3	TC4	TC5
L2 -> L1D	1925	1925	1925	3347	3347	3347
L2 -> L2	2028	2028	2028	3547	3547	3547
L2 -> DDR2	2086	2086	2086	2580	2580	2580
DDR2 -> L2	2087	2074	2087	2026	2026	2026
DDR2 -> DDR2	983	984	983	974	976	975

In conclusion, TC3, TC4, and TC5 achieve about two times the bandwidth as TC0, TC1 and TC2. Without special note, all performance data in this application report is measured on TC3.

4.3 EDMA Bandwidth vs. Transfer Flexibility

EDMA3 channel parameters allow many different transfer configurations. Most typical transfers burst properly, and memory bandwidth is fully utilized. However, in some less common configurations, transfers are unable to burst, reducing performance. To properly design a system, it is important to know which configurations offer the best performance for high speed operations, and which must trade throughput for flexibility.

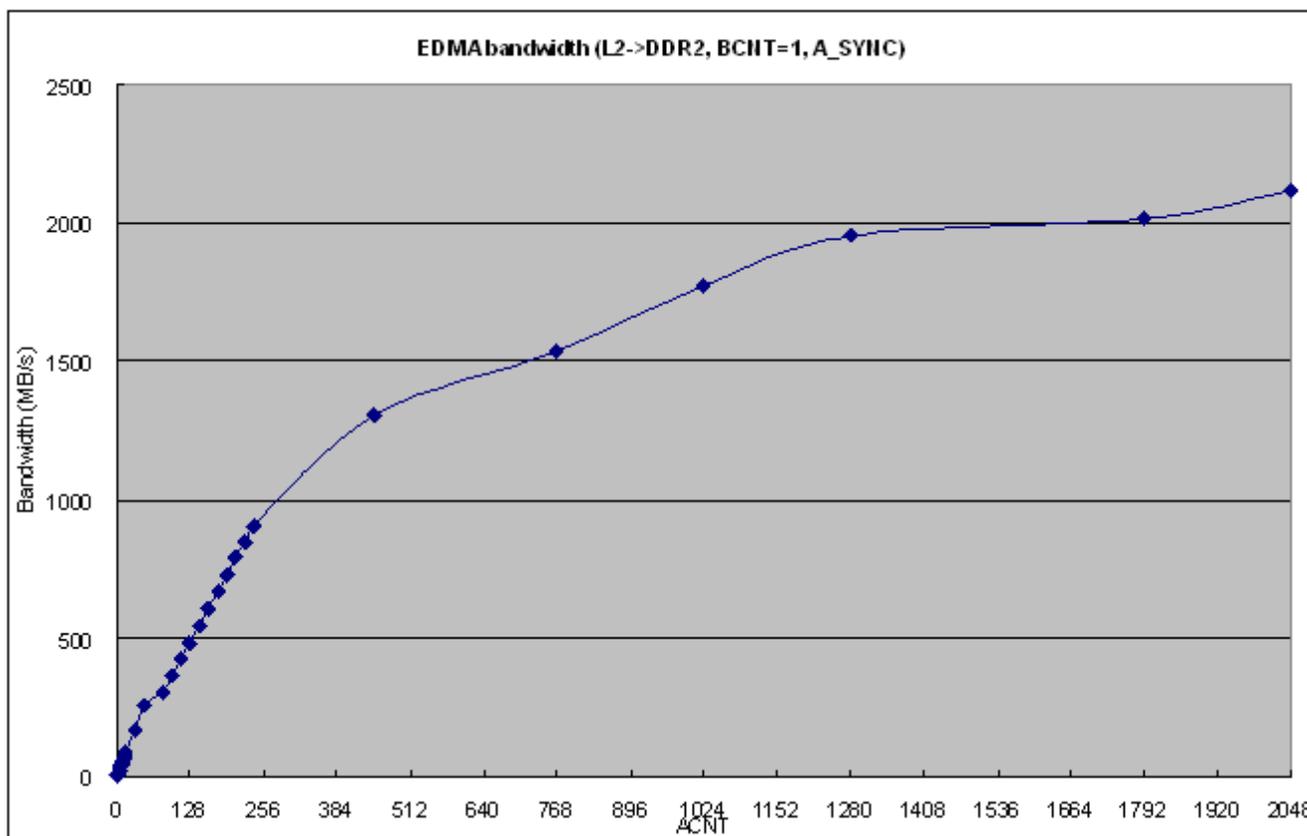
4.3.1 First Dimension Size (ACNT) Considerations, Burst Width

To make full utilization of bandwidth in the transfer engine, it is important to fully utilize the bus width available and allow for data bursting.

ACNT size should be a multiple of 16 bytes to fully utilize the 128-bit bus width. ACNT should be a multiple of 64 bytes to fully utilize the 64-byte default burst width. ACNT should be a multiple of 256 bytes to fully utilize the 256-byte FIFO.

Figure 6 shows performance data from a TMS320TCI6488 running at 1GHz, transferring 1~2048 bytes from L2 to DDR2 using an EDMA3 channel.

Figure 6 Effect of ACNT Size on EDMA Bandwidth



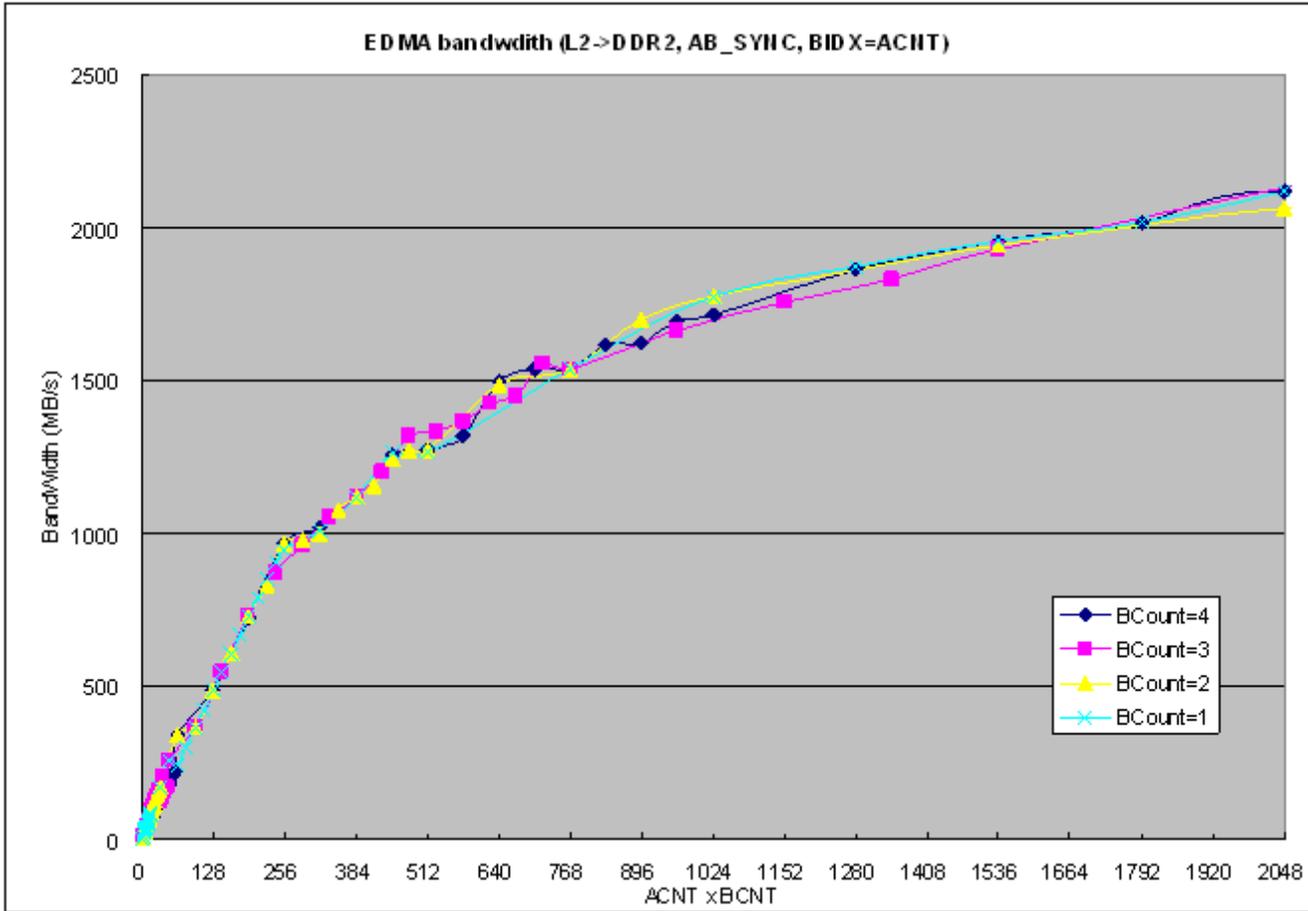
In conclusion, the bigger the ACNT, the more bandwidth can be achieved.

4.3.2 Two Dimension Considerations, Transfer Optimization

If 2D transfer (AB_Sync) is linear (BIDX=ACNT), the 2D transfer will be optimized as a 1D transfer.

Various ACNT and BCNT combinations were investigated; however, the overall transfer size (ACNT * BCNT) was proved to have more bearing than the particular combination settings. Figure 7 shows the linear 2D transfer test result. It shows that no matter what BCNT, the bandwidths are similar as long as ACNTxBCNT are same.

Figure 7 Linear 2D Transfer



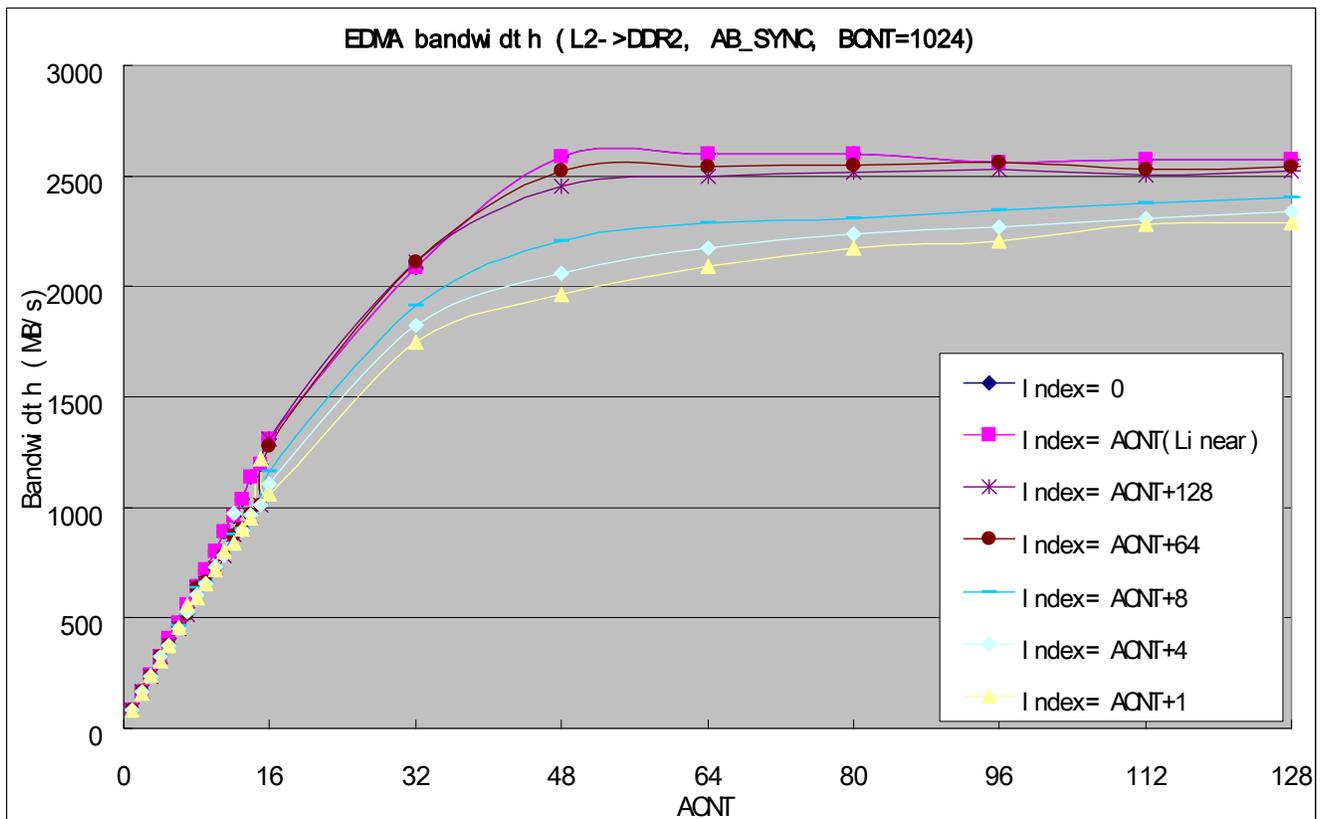
If 2D transfer is not linear, the bandwidth utilization is determined by the ACNT as shown in Figure 6.

4.3.3 Index Consideration

Index dramatically affects the EDMA throughput. Linear transfer (Index= ACNT) fully utilizes bandwidth. Fixed Index (Index= 0) can utilize the same bandwidth as linear transfer. Other index modes will lower the EDMA performance. Odd index has the worst performance. If index is power of 2, and it is larger than 8, the performance degradation is very small.

Figure 8 shows the index effect on EDMA throughput, transferring 1024 rows (BCNT= 1024) of 2D data from L2 to DDR2, with different index.

Figure 8 Index Effect on EDMA Transfer



Without special note, all performance data in this application report is measured with Index= 0 or Index= ACNT.

4.3.4 Address Alignment

Address alignment may slightly impact the performance. The default burst size of EDMA3 is 64bytes. If the transfer is across the 64-byte boundary, then the EDMA3 TC breaks the ACNT array into 64-bytes burst to the source/destination addresses. So, if the source or destination address is not aligned to the 64-byte boundary, and the transfer is across the 64-byte boundary, an extra burst will be generated to handle the unaligned head and tail data.

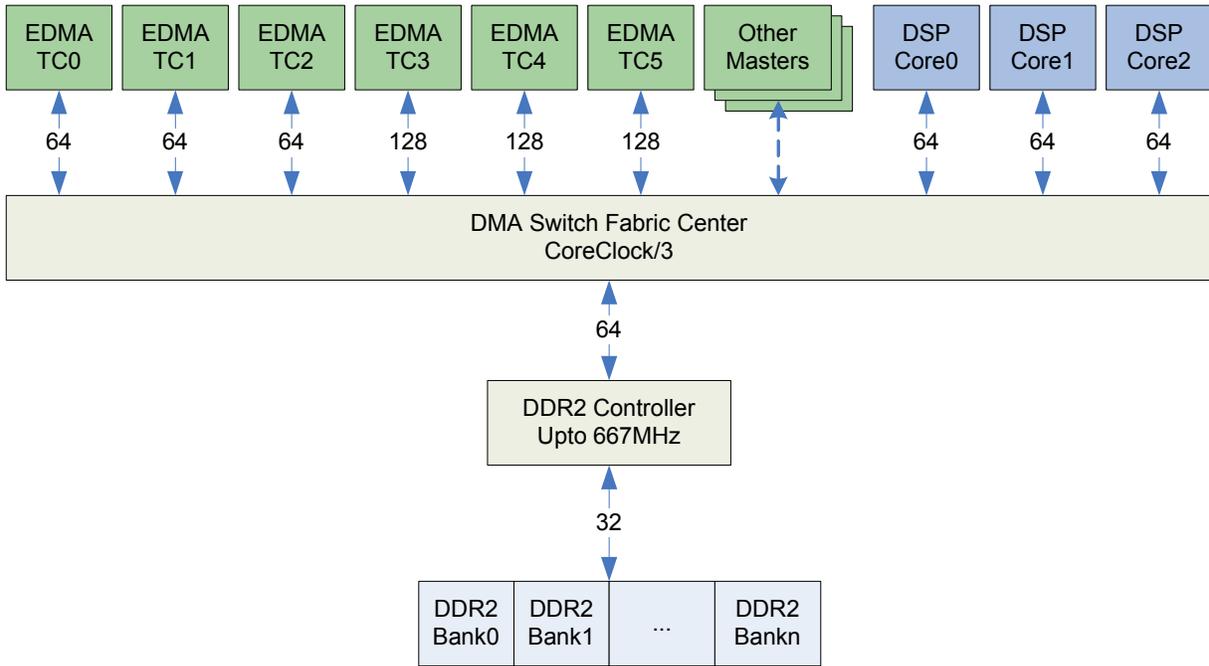
For big transfers, this overhead may be ignored. All data presented in this document are based on the address aligned transfer.

5 Performance of Multiple Masters Sharing DDR2

Since the TCI6488 includes three cores and multiple DMA masters, they may access DDR2 memory in parallel. This section discusses the performance of multiple masters sharing DDR2 memory.

Figure 9 shows the DDR2 access paths.

Figure 9 DDR2 architecture and access paths



Multiple masters access the DDR2 SDRAM through the DMA switch fabric center. If multiple masters access it at same time, it is arbitrated on the DDR2 controller based on priority of the masters.

The bank number of different DDR2 devices may be different. The DDR2 controller on the TCI6488 can support DDR2 memory with 1, 2, 4, or 8 banks. The DDR2 on the TCI6488 EVM has 8 banks. The data on it is organized as described below. Please note that the row size for different DDR2 devices may be different as well.

Table 8 Data Organization on DDR2 Banks

	Bank 0	Bank 1	...	Bank 7
Row 0	byte 0~4095	byte 4096~8191		byte 4096*7~4096*8-1
Row 1	byte 4096*9~4096*9-1	byte 4096*9~4096*10-1		byte 4096*15~4096*16-1
...

Although DDR2 has multiple banks, there are no multiple buses to connect to it. So, the bank number does not directly improve the throughput.

The DDR2 SDRAM is accessed based on row. Before master access data in a row, it must open the row first, and then it can randomly access any bytes in the row. If the master wants to access data in a new row in same bank, the old row must be closed first, and then open the new row in the same bank for access. The row switch (row close/open) operations introduce extra delay cycles, which is called row switch overhead.

Every bank can have one open row, so the DDR2 with eight banks can have eight open rows at the same time, which dramatically reduces the probability of row switch. For example, after a master opens row 0 in bank 0 for access, it can open row 1 in bank 1 without closing the row 0 in bank 0, and then the master can access both row 0 in bank 0 and row 1 in bank 1 randomly without row switch overhead.

Two data structures for testing are defined to verify the effect of DDR2 row switch (row close/open) overhead.

Table 9 Multiple Master Access Different Rows on Same DDR2 Bank

	Bank 0	Bank 1	Bank 2	...	Bank n	...
Row 0	Master 0 Access Range					
Row 1	Master 1 Access Range					
Row 2	Master 2 Access Range					
...	...					
Row n	Master n Access Range					
...	...					

The above case is the worst case with maximum row switch overhead. Every master access may result in a row switch.

The following case is the best case without any row switch because every master always accesses an open row dedicated for it.

Table 10 Multiple Master Access Different Rows on Different DDR2 Banks

	Bank 0	Bank 1	Bank 2	...	Bank n	...
Row 0	Master 0 Access Range					
Row 1		Master 1 Access Range				
Row 2			Master 2 Access Range			

Table 10 Multiple Master Access Different Rows on Different DDR2 Banks

	Bank 0	Bank 1	Bank 2	...	Bank n	...
...	...					
Row n					Master n Access Range	
...	...					

5.1 Performance of Multiple DSP Cores Sharing DDR2

Table 11 shows the performance of multiple DSP cores sharing the 32-bit 667MHz DDR2 on the 1GHz TCI6488 EVM under different scenarios. The DDR2 for test is cacheable, the L1D cache is set to 32KB, and the L2 cache size is set to 256KB. The non-cacheable case is not measured because it demands less bandwidth than the cacheable case. The priority for all DSP cores is the same for this test.

Table 11 Performance of Multiple DSP Cores Sharing DDR2

L2->DDR2, different row on different DDR2 bank				L2->DDR2, different row on same DDR2 bank			
Master	Bandwidth (MB/s)			Master	Bandwidth (MB/s)		
Core 0	443	445	445	Core 0	443	403	404
Core 1		445	445	Core 1		404	404
Core 2			445	Core 2			404
Total Bandwidth	443	890	1335	Total Bandwidth	443	807	1212
DDR2->L2, different row on different DDR2 bank				DDR2->L2, different row on same DDR2 bank			
Master	Bandwidth (MB/s)			Master	Bandwidth (MB/s)		
Core 0	582	579	582	Core 0	582	515	463
Core 1		580	583	Core 1		518	464
Core 2			583	Core 2			466
Total Bandwidth	582	1159	1748	Total Bandwidth	582	1033	1393

The above table shows the DDR2 has enough bandwidth to support multiple DSP cores accessing DDR2 simultaneously. When three cores access the DDR2 simultaneously, the total bandwidth required is less than the theoretical bandwidth of DDR2 (2667MB/s). The throughput limitation is on the master itself instead of the DDR2.

The performance of multiple cores accessing different rows on the **same** DDR2 bank is worse than the performance of multiple cores accessing different rows on **different** DDR2 banks, and the reason is the DDR2 row switch overhead.

5.2 Performance of Multiple EDMA Sharing DDR2

Table 12 shows the performance of multiple EDMA TCs sharing DDR2 measured on 32-bit 667MHz DDR2 on the 1GHz TCI6488 EVM under different conditions. The priority for all EDMA TCs is the same for this test.

Table 12 Performance of Multiple EDMA Sharing DDR2

L2->DDR2, same priority, different row on different DDR2 bank, BPRIO=0x7F												
Master	Bandwidth(MB/s)											
EDMA TC0	2086	1289	856	642	514	514						514
EDMA TC1		1289	856	642	514	514					642	514
EDMA TC2			856	642	514	514				859	642	514
EDMA TC3				642	514	514			1289	859	642	514
EDMA TC4					514	257		1290	645	429	322	257
EDMA TC5						257	2580	1290	645	429	322	257
Total Bandwidth	2086	2578	2568	2568	2570	2570	2580	2580	2579	2576	2570	2570
L2->DDR2, same priority, different row on same DDR2 bank, BPRIO=0x7F												
Master	Bandwidth(MB/s)											
EDMA TC0	2086	383	255	191	171	171						171
EDMA TC1		383	255	191	170	171					220	171
EDMA TC2			255	191	169	170				334	221	170
EDMA TC3				191	169	170			1289	334	220	170
EDMA TC4					168	85		1290	645	167	110	85
EDMA TC5						85	2580	1290	645	167	110	85
Total Bandwidth	2086	766	765	764	847	852	2580	2580	2579	1002	881	852
DDR2->L2, same priority, different row on different DDR2 bank, BPRIO=0x7F												
Master	Bandwidth(MB/s)											
EDMA TC0	2087	1281	851	639	512	512						512
EDMA TC1		1281	851	639	512	512					640	512
EDMA TC2			851	639	512	512				733	640	512
EDMA TC3				639	512	512			854	733	693	512
EDMA TC4					512	256		1278	854	548	381	256
EDMA TC5						256	2026	1278	854	548	379	256
Total Bandwidth	2087	2562	2553	2556	2560	2560	2026	2556	2562	2562	2733	2560
DDR2->L2, same priority, different row on same DDR2 bank, BPRIO=0x7F												
Master	Bandwidth(MB/s)											
EDMA TC0	2074	959	639	479	377	377						377
EDMA TC1		960	639	479	377	377					447	377
EDMA TC2			639	479	377	377				320	446	377
EDMA TC3				479	377	377			849	789	558	377
EDMA TC4					377	188		1277	845	624	338	188
EDMA TC5						188	2026	1277	845	624	338	188
Total Bandwidth	2074	1919	1917	1916	1885	1884	2026	2554	2539	2357	2127	1884

The above table shows the bandwidth of DDR2 is not enough to support multiple EDMA TCs accessing it simultaneously. The available bandwidth is equally split between EDMA TCs. Please note that TC4 and TC5 share the same bridge (Br5) to DDR2 (refer to Section 4 - System Interconnect of the TCI6488 data sheet), so the sum of their bandwidths is close to the bandwidth of other TCs.

5.3 Effect of DDR2 Bank and Row Switch on Multiple Masters Sharing DDR2

The performance of multiple EDMA TCs accessing different rows on the **same** DDR2 bank is much worse than the performance of multiple EDMA TCs accessing different rows on **different** DDR2 banks, and the reason is the DDR2 row switch overhead. The result becomes worse when the DDR2 load becomes heavy. The worst case is multiple EDMA TCs writing to different rows on same DDR2 bank, which is almost dominated by the row switch overhead because every write burst results in row switch.

The probability of multiple masters accessing the **same** DDR2 bank depends on the master number and DDR2 bank number. For example, if four EDMA randomly access DDR2 memory, the probability of at least two TCs accessing the same DDR2 bank is:

$$1 - C_8^4 P_4^4 / 8^4 = 59\%$$

[Table 13](#) lists the probability for different combinations of master number and bank number.

Table 13 Probability of Multiple Masters Access Same DDR2 Bank

	2 Masters	4 Masters	6 Masters	8 Masters	10 Masters
4 Banks	25%	90.6%	100%	100%	100%
8 Banks	12.5%	59%	92.3%	99.7%	100%

According to the above data, to reduce the row switch overhead, DDR2 memory with eight banks is strongly recommended.

The DDR2 controller on a TCI6488 is optimized to alleviate the row switch overhead. The DDR2 controller makes use of the following FIFOs:

- Write Command FIFO - stores up to seven write commands from masters
- Write Data FIFO - stores up to 176 bytes of data from masters
- Read Command FIFO - stores up to 22 read commands from masters
- Read Data FIFO - stores up to 272 bytes of data to masters

The DDR2 memory controller performs command re-ordering and scheduling in an attempt to achieve efficient transfers with maximum throughput. The goal is to maximize the utilization of open rows, while hiding the overhead of opening and closing DDR2 SDRAM rows. Command re-ordering takes place within the command FIFOs.

The DDR2 memory controller examines each of the commands in the FIFOs and performs the following reordering:

- Among all pending reads, selects reads to rows already open. Among all pending writes, selects writes to rows already open.
- Selects the highest priority command from pending reads and writes to open rows. If multiple commands have the highest priority, then the DDR2 memory controller selects the oldest command.
- If the Read FIFO is not full, then the read command will be performed before the write command, otherwise the write command will be performed first.

As the above test data shows, the read performance degradation caused by row switch is much less than the write performance degradation caused by row switch. The reason is that the read FIFOs are much bigger than the write FIFOs.

Another method that may alleviate the row switch overhead is to organized data for different masters as follows.

Table 14 Organize Data Buffer in Different Banks to Alleviate Row Switch Overhead

	Bank 0	Bank 1	Bank 2	...	Bank n	...
Row 0	Data buffer for master 0	Data buffer for master 1	Data buffer for master 2	...	Data buffer for master n	...
Row 1						
Row 2						
...						
Row n						
Row n+1
...

A special EDMA configuration ($BIDX = (\text{row size}) * (\text{bank number})$) can be utilized to transfer the above data between the L2 memory and DDR2 memory.

5.4 Effect of Priority on Multiple Masters Sharing DDR2

If there are multiple commands access to open rows, or all commands do not access to open rows, the DDR2 controller arbitrates multiple requests further according to the priority of the masters. All of the above data are measured with the same priority for all masters. Table 15 shows the effect of priority when multiple EDMA TCs access to DDR2 in parallel.

Table 15 Effect of Priority on Multiple EDMA TCs Access to DDR2

DDR2->L2, different priority, different row on different DDR2 bank, BPRI0=0x7F														
Master	Priority	Bandwidth (MB/s)						Priority	Bandwidth (MB/s)					
EDMA TC0	0	2087	2051	1580	1280	1280	1280	7						0
EDMA TC1	1		511	906	993	993	993	6						0
EDMA TC2	2			75	224	224	224	5				68	68	68
EDMA TC3	3				64	64	64	4			75	239	239	238
EDMA TC4	4					0	0	3		642	904	1007	1007	1007
EDMA TC5	5						0	2	2026	1920	1582	1247	1246	1247
Total Bandwidth		2087	2562	2561	2561	2561	2561		2026	2562	2561	2561	2560	2560

The above data prove that the priority dramatically affects the bandwidth allocation between multiple read operations when the DDR2 is heavily loaded.

Additional testing shows that the priority does not affect bandwidth allocation for EDMA write operations because it is dominated by row switch overhead. For multiple DSP cores accessing the DDR2, the priority does not affect the bandwidth allocation between DSP cores because three DSP cores cannot make the DDR2 load heavy.

The reordering and scheduling rules listed above may lead to command starvation (bandwidth= 0), which is the prevention of certain commands from being processed by the DDR2 memory controller. A continuous stream of DDR2 SDRAM commands to a row in an open bank can block commands to the closed row in the same bank.

To avoid these conditions, the DDR2 memory controller can momentarily raise the priority of the oldest command in the command FIFO after a set number of transfers have been made. The PRIO_RAISE field in the Burst Priority Register (BPRIO) sets the number of the transfers that must be made before the DDR2 memory controller will raise the priority of the oldest command. Leaving the PRIO_RAISE bits at their default value (0xFF) disables this feature of the DDR2 memory controller. This means commands can stay in the command FIFO indefinitely. Therefore, these bits should be set to another value immediately following reset to enable this feature according to application requirements.

All of the above data are measured with BPRIO= 0x7F. The following table shows the effect of the BPRIO register.

Table 16 Effect of BPRIO on Multiple EDMA TCs Access to DDR2

DDR2->L2, different priority, different row on different DDR2 bank													
Master	Priority	BPRIO=0xFF				BPRIO=0x7F				BPRIO=0			
EDMA TC0	0	1601	1282	1281	1281	1580	1280	1280	1280	1278	1165	1164	1164
EDMA TC1	1	961	1025	1025	1025	906	993	993	993	640	559	559	559
EDMA TC2	2	0	254	254	254	75	224	224	224	640	558	557	557
EDMA TC3	3		0	0	0		64	64	64		278	278	278
EDMA TC4	4			0	0			0	0			0	0
EDMA TC5	5				0				0				0
Total Bandwidth		2562	2561	2560	2560	2561	2561	2561	2561	2558	2560	2558	2558

The above data shows more masters are starved (bandwidth= 0) when BPRIO=0xFF. The bandwidth allocation is much more fair with BPRIO=0.

6 References

1. TMS320TCI6488 DSP Enhanced DMA (EDMA3) Controller User's Guide ([SPRUEE9](#))
2. TMS320C64x+ Megamodule Reference Guide ([SPRU871](#))
3. TMS320TCI6488 datasheet ([SPRS300](#))
4. TMS320TCI6482 EDMA3 Performance ([SPRAAG8](#))

IMPORTANT NOTICE

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

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

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

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

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

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

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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

Products

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

Applications

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

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2009, Texas Instruments Incorporated