

TMS320DM6467 SoC Architecture and Throughput Overview

DSPS Applications

ABSTRACT

This application report provides information on the DM6467 throughput performance and describes the DM6467 System-on-Chip (SoC) architecture, data path infrastructure, and constraints that affect the throughput and different optimization techniques for optimum system performance. This document also provides information on the maximum possible throughput performance of different peripherals on the SoC.

Contents

1	SoC Architectural Overview	. 2
2	SoC Constraints	10
3	SoC Level Optimizations	12
	IP Throughput Optimization Techniques	
5	References	46
Appen	dix A EDMA High-Resolution Diagrams	47

List of Figures

1	TMS320DM6467 System Interconnect Block Diagram	. 3
2	TMS320DM6467 Peripheral Configuration Bus	. 4
3	Concurrent Transactions Through SCR	. 7
4	Bridge	. 7
5	Bus-Width and Clock Rate Conversion	. 8
6	EMAC-to-DDR Transfer	10
7	Bridge Head of Line Blocking	11
8	EDMA3 Controller Block Diagram	14
9	EDMA3 Channel Controller (EDMA3CC) Block Diagram	15
10	EDMA3 Transfer Controller (EDMA3TC) Block Diagram	16
11	Utilization of EDMA for L2, DDR Access	18
12	Utilization for Different Element Size (ACNT)	19
13	Effect of A-Sync and AB-Sync	20
14	Utilization for Different Destination Index Value	21
15	Performance of TC0 and TC1	22
16	Utilization for Different Burst Size Configuration	23
17	Utilization for Different Source and Destination Alignment	24
18	Utilization for EDMA for Different CPU and DDR Frequency	25
19	EDMA Performance	26
20	16- and 32-Bit Element Throughput Analysis	29
21	8-Bit Element Throughput Analysis	30
22	Throughput Dependency on SRC/DST Buffer Location (TX Queue = 1, RX Queue = 1, TX Trigger Level = 20, RX Trigger Level = 60)	33
23	Throughput Dependency on TX/RX FIFO Trigger Level (TX Queue = 0, RX Queue = 0, TX Buffer in AEMIF Memory, RX Buffer in DDR Memory)	33
24	EMAC and MDIO Block Diagram	35
25	Ethernet Frame Format	

1



26	Effect of Packet Size on the EMAC Throughput for 100 Mbps Mode	37
27	Effect of Packet Size on the EMAC Throughput for Giga Bit Mode	38
28	Effect of Descriptor Memory Location on the EMAC Throughput for 100 Mbps mode	39
29	Effect of Descriptor Memory Location on the EMAC Throughput for Giga Bit Mode	40
30	Effect of Source Memory Location on the EMAC Throughput for 100 Mbps Mode	41
31	Effect of Source Memory Location on the EMAC Throughput for Giga Bit mode	42
32	Effect of Destination Memory Location on the EMAC Throughput for 100 Mbps Mode .	43
33	Effect of Destination Memory Location on the EMAC Throughput for Giga Bit Mode	44
34	Effect of Different Memory Locations on the EMAC Throughput	45
A-1	Utiliztation of EDMA for L2, DDR Access	
A-2	Utilization for Different Element Size (ACNT)	
A-3	Effect of A-Sync and AB-Sync	49
A-4	Utilization for Different Destination Index Value	50
A-5	Performance of TC0 and TC1	51
A-6	Utilization for Different Burst Size Configuration	52
A-7	Utilization for Different Source and Destination Alignment	53
A-8	Utilization for EDMA for Different CPU and DDR Frequency	54
A-9	EDMA Performance	55

List of Tables

1	TMS320DM646x DMSoC Master Peripherals	. 4
2	TMS320DM646x DMSoC Slaves	. 5
3	System Connection Matrix	
4	Memory Maximum Bandwidths	12
5	Default Master Priorities	12
6	Frequency and Bus Widths for Different Memory and Slave Endpoints	16
7	Factors Considered for Throughput	17
8	Read/Write Command Optimization Rules	21
9	EDMA3 Transfer Controller Configurations	22
10	Performance of EDMA for 8KB or 16KB Transfer	26
11	Factors Affecting McASP Throughput	28
12	UART Modem Mode Baud Rate	31
13	UART IrDA Mode Baud Rate	31
14	Possible Effective Factors of UART Throughput	32
15	Ethernet Frame Description	36
16	Factors Considered for Throughput	37
17	Effect of Different Memory on the EMAC Throughput	46

1 SoC Architectural Overview

Figure 1 and Figure 2 show that in the DM6467 SoC, the C64x+[™] megamodule, the ARM subsystem, the enhanced direct memory access (EDMA3) transfer controllers (TC), and the system peripherals are interconnected through a switch fabric architecture. The switch fabric is composed of multiple switched central resources (SCRs) and multiple bridges. More information on SCR and bridges is provided later in this document.

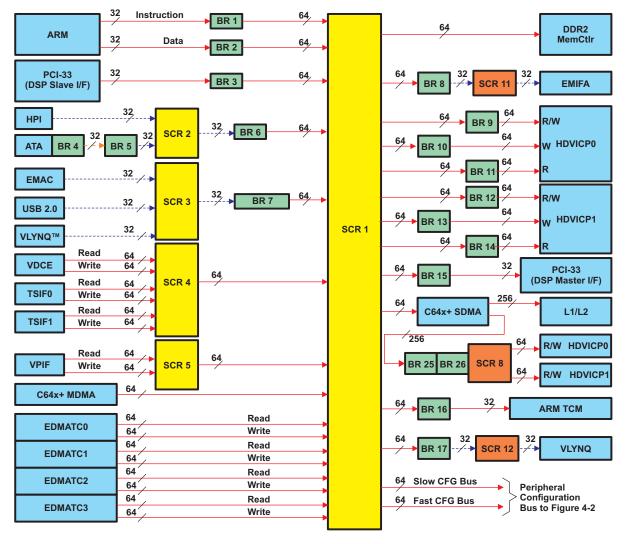
The following is a list of points that help to interpret Figure 1 and Figure 2.

- The arrow indicates the master/slave relationship.
- The arrow originates at a bus master and terminates at a bus slave.
- The direction of the arrows does not indicate the direction of data flow. Data flow is typically bi-directional for each of the documented bus paths.

C64x+, VLYNQ are trademarks of Texas Instruments. All other trademarks are the property of their respective owners.



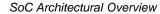
- The pattern of each arrow's line indicates the clock rate at which it is operating, either DSP/2 or DSP/4 clock rate.
- Some peripherals may have multiple instances shown for a variety of reasons in the diagrams, some of which are described below:
 - The peripheral/module has master port(s) for data transfers, as well as slave port(s) for register access, data access, and/or memory access. Examples of these peripherals are C64x+ megamodule, EDMA3, AT attachment (ATA), universal serial bus (USB), Ethernet Media Access Controller (EMAC), Video Port Interface (VPIF), VLYNQ[™], and Host Port Interface (HPI).



DSP/2 Clock Rate
 DSP/4 Clock Rate
 OTHER

Figure 1. TMS320DM6467 System Interconnect Block Diagram





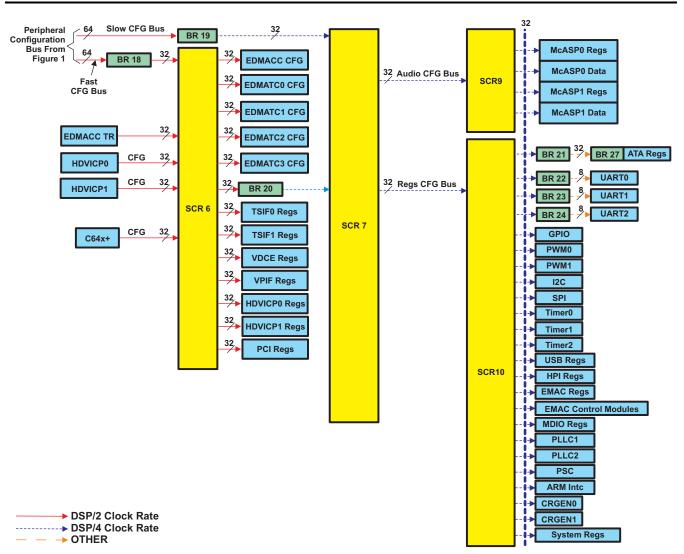


Figure 2. TMS320DM6467 Peripheral Configuration Bus

1.1 Master Peripherals

4

The DM6467 SoC peripherals can be classified into two categories: master peripherals and slave peripherals. Master peripherals are typically capable of initiating read and write transfers in the system and do not rely on the EDMA3 (system DMA) or CPU to perform transfers to and from them.

Table 1 lists all master peripherals of the DM6467 SoC. To determine the allowed connections between masters and slaves, each master request source must have a unique master ID (mstid) associated with it. The master ID for each DM6467 SoC master is also shown in Table 1.

Mstid	Master
0	ARM Instruction
1	ARM Data
2	C64x+ MDMA
3	C64x+ CFG
4-7	Reserved

Table 1. TMS320DM646x DMSoC Master Peripherals

Mstid	Master
8	HDVICP0 CFG
9	HDVICP1 CFG
10	EDMA CC TR
11-15	Reserved
16	EDMA TC0 Read Port
17	EDMA TC0 Write Port
18	EDMA TC1 Read Port
19	EDMA TC1 Write Port
20	EDMA TC2 Read Port
21	EDMA TC2 Write Port
22	EDMA TC3 Read Port
23	EDMA TC3 Write Port
24-31	Reserved
32	PCI
33	HPI
34	ATA
35	EMAC
36	USB
37	VLYNQ
38	VPIF mstr1 Read Port
39	VPIF mstr0 Write Port
40	TSIF0 Read Port
41	TSIF0 Write Port
42	TSIF1 Read Port
43	TSIF1 Write Port
44	VDCE Write Port
45	VDCE Read Port
46-63	Reserved

Table 1. TMS320DM646x DMSoC Master Peripherals (continued)

1.2 Slave Peripherals

Slave peripherals service the read/write transactions that are issued by master peripherals. All DM6467 SoC slaves are listed in Table 2. Note that memories are also classified as peripherals.

Table 2. TMS320DM646x DMSoC Slaves

Slaves		
DDR2 Memory Controller		
EMIFA		
HDVICP0/1 Read Port		
HDVICP0/1 Write Port		
HDVICP0/1 Read/Write Port		
PCI Slave		
C64x+ SDMA		
ARM TCM		
VLYNQ Slave		
VLYNQ Regs		
EDMA3CC Regs		

SoC Architectural Overview



www.ti.com

	•	
Slaves		
EDMA3TC0/1/2/3 Regs		
TSIF0/1 Regs		
VDCE Regs		
VPIF Regs		
HDVICP0/1 Regs		
PCI Regs		
McASP0/1 Regs		
McASP0/1 Data		
ATA Regs		
UART0/1/2		
General-Purpose Input/Output (GPIO)		
PWM0/1		
I2C		
SPI		
TIMER0/1/2		
USB Regs		
EMAC Regs		
EMAC Control Modules		
MDIO Regs		
PLLC1/2		
PSC		
ARM INTC		
CRGEN0/1		
SYSTEM Regs		

Table 2. TMS320DM646x DMSoC Slaves (continued)



1.3 Switched Central Resources (SCR)

The SCR is an interconnect system that provides low-latency connectivity between master peripherals and slave peripherals (more information on master and slave peripherals is provided later in this document). A SCR is the decoding, routing, and arbitration logic that enables the connection between multiple masters and slaves that are connected to it. As shown in Figure 1 and Figure 2, multiple SCRs are used in the DM6467 SoC to provide connections among different peripherals. Refer to Table 3 for supported master and slave peripheral connections. Additionally, the SCRs provide priority-based arbitration and facilitate concurrent data movement between master and slave peripherals. For example, as shown in Figure 3 (black lines), through SCR1, the ARM data (master) can send data to the DDR2 memory controller (slave) concurrently without affecting a data transfer between the EMAC (master) and L2 memory (slave).

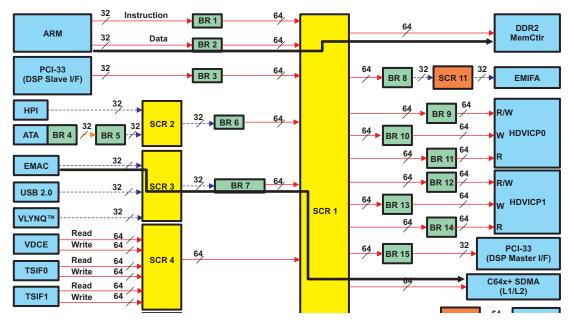
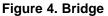


Figure 3. Concurrent Transactions Through SCR

1.4 Bridge

In the DM6467 SoC, different clock rates and bus widths are used in various parts of the system. To communicate between two peripherals that are operating at different clock rates and bus widths, logic is needed to resolve these differences. Bridges provide a means of resolving these differences by performing bus-width conversion as well as bus operating clock frequency conversion. Bridges are also responsible for buffering read and write commands and data. Figure 4 shows the typical connection of a bridge.







SoC Architectural Overview

www.ti.com

Multiple bridges are used in the DM6467 SoC. For example, as shown in Figure 5, Bridge 1 (BR1) performs a bus-width conversion between a 32-bit bus and a 64-bit bus. Also, Bridge 8 (BR8) performs a frequency conversion between a bus operating at DSP/2 clock rate and a bus operating at DSP/4 clock rate along with a bus-width conversion between a 64-bit bus and a 32-bit bus.

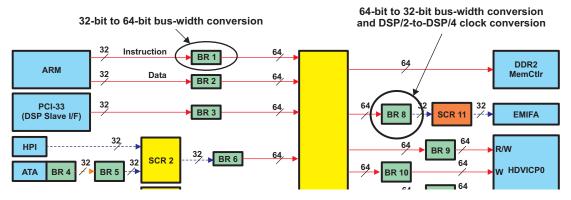


Figure 5. Bus-Width and Clock Rate Conversion



1.5 Master/Slave Connectivity

Not all master peripherals can connect to all slave peripherals. The supported master and slave connections are designated by an X in Table 3.

												SLA	٩VE											
																F	FAST	CFG	REGS	3				
MASTER	D D R	EMIFA	н ם > - с ь о ӄ Я	Зо до-<01	H D V I C P O (R)	ЗЭ → О − < О Н	H D V I C P 1 ()	H D V I C P 1 (R)	P C I S L A V E	C 6 4 + S D M A	A R M T C M	V L Y N Q S L A V E	V L Y N Q R E G S	E D M A 3 C C	E D M A 3 T C 0	E D M A 3 T C 1	E D M A 3 T C 2	E D M A 3 T C 3	T S I F V D C E V P I F	H D V I C P O R E G S	H D V I C P 1 R E G S	P C I R E G S	SLOWCFGREGS	A U D I O C F G R E G S
ARM IP	Х	Х	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ARM DP	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	-	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х
C64x+ MDMA	х	х	х	-	-	х	-	-	х	-	х	х	х	-	-	-	-	-	-	-	-	-	-	-
C64x+ CFG	-	-	-	-	-	1	-	-	-	-	-	-	Ι	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х
HDVICP0 CFG	-	-	-	-	-	-	-	-	-	-	-	-	-	х	х	х	Х	х	-	-	Х	-	Х	-
HDVICP1 CFG	-	-	-	-	-	-	-	-	-	-	-	-	-	х	х	х	х	х	-	х	-	-	Х	-
PCI	Х	-	-	-	-	-	-	-	-	Х	Х	Х	-	-	-	-	-	-	-	-	-	-	Х	Х
HPI	Х	-	-	-	-	-	-	-	-	Х	Х	Х	-	-	-	-	-	-	-	-	-	-	Х	Х
ATA	Х	-	-	-	-	-	-	-	-	Х	Х	Х	-	-	-	-	-	-	-	-	-	-	-	-
EMAC	X	-	-	-	-	-	-	-	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-
USB	X	X	-	-	-	-	-	-	-	X	X	Х	-	-	-	-	-	-	-	-	-	-	-	-
VLYNQ VPIF Write	X X	× _	X -	-	-	Х	-	-	-	X –	X _	-	-	-	-	-	-	-	-	-	-	-	X -	Х
VPIF Read	X	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
VDCE Read	Х	Х	-	_	-	_	-	_	_	_	Х	_	-	_	_	_	-	_	_	-	-	_	-	-
VDCE Write	Х	Х	-	-	-	-	-	-	-	-	Х	-	-	-	-	-	-	-	-	-	-	-	-	-
TSIF0 Read	Х	Х	-	-	-	-	-	-	-	-	Х	Х	-	-	-	-	-	-	-	-	-	-	-	-
TSIF0 Write	Х	Х	I	-	I	-	I	-	-	-	Х	Х	I	-	-	-	I	-	-	I	I	-	I	-
TSIF1 Read	Х	Х	-	-	-	-	-	-	-	-	Х	Х	-	-	-	-	-	-	-	-	-	-	-	-
TSIF1 Write	Х	Х	-	-	-	-	-	-	-	-	Х	Х	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	х	х	Х	х	-	-	-	-	-	-
EDMA3TC0 Read	Х	Х	Х	-	Х	Х	-	Х	х	х	Х	Х	-	х	-	-	-	-	Х	Х	Х	х	Х	х
EDMA3TC0 Write	х	х	х	х	-	х	Х	-	х	х	х	х	-	х	-	-	-	-	х	х	х	х	Х	х
EDMA3TC1 Read	х	х	х	-	х	х	-	х	х	х	х	х	-	х	-	-	-	-	х	х	х	х	х	х
EDMA3TC1 Write	х	х	х	х	I	х	х	-	х	х	х	х	I	х	-	-	I	-	х	х	х	х	х	х
EDMA3TC2 Read	х	х	х	-	х	х	I	х	х	х	х	х	I	х	-	_	I	_	х	х	х	х	х	х
EDMA3TC2 Write	х	х	х	х	-	х	х	-	х	х	х	х	-	х	-	-	-	-	х	х	х	х	х	х
EDMA3TC3 Read	х	х	х	-	х	х	-	х	х	х	х	х	-	х	-	-	-	-	х	х	х	х	х	х
EDMA3TC3 Write	х	х	х	х	-	х	х	-	х	х	х	х	-	х	-	-	-	-	х	х	х	х	х	х

Table 3. System Connection Matrix⁽¹⁾

⁽¹⁾ "X" denotes supported connections and "-" denotes unsupported connections.



1.6 Data Bus (widths/speeds)

There are two main types of busses on the DM6467 SoC:

- A 64-bit bus with separate read and write interface, allowing multiple outstanding read and write transactions simultaneously. This bus is best suited for high-speed/high-bandwidth exchanges, especially data transfers between on-chip and off-chip memories. On the DM6467 device, the main SCR (SCR1) interfaces with all the modules using this 64-bit bus. Most of the high bandwidth master peripherals (e.g., EDMA3TC) and slave memories (e.g., C64x+ system direct memory access (SDMA) port for L1/L2 memory access, DDR2, etc.) are directly connected to the main SCR through this 64-bit bus. Peripherals that do not support the 64-bit bus interface are connected to the main SCR via bridges (responsible for protocol conversion from 64-bit to 32-bit bus interface).
- A 32-bit bus, with a single interface for both reads and writes. The read and write transactions are serviced strictly in order. This bus is best suited for communication with the memory-mapped registers of all on-chip peripherals. Accesses to memory-mapped registers could be for configuration purposes (e.g., accesses to configure a peripheral) or for data accesses (e.g., read writes from/to multichannel audio serial port (McASP) receive/transmit buffer registers or writes to transmitter holding registers and reads from receiver buffer registers on universal asynchronous receive/transmitter (UART)).

1.7 Default Burst Size

Burst size is another factor that affects peripheral throughput. A master's read/write transaction is broken down into smaller bursts at infrastructure level. The default burst size for a given peripheral (master) is the maximum number of bytes per read/write command. The burst size determines the intra-packet efficiency of a master's transfer. At system interconnect level, it also facilitates *pre-emption* as the SCR arbitrates at burst size boundaries. See Section 4.1.3.5 for more details on how the burst size affects the performance of EDMA transfers.

2 SoC Constraints

This section describes the factors that constrain the system throughput.

2.1 HW Latency

Each master-slave transaction has to go through multiple elements in the system. Each element contributes to a hardware latency of the transaction. In Figure 1 and Figure 2, all masters, slave, SCRs and bridges contribute latency.

For example, consider a transfer from EMAC-to-DDR memory as shown in Figure 6 (black line).

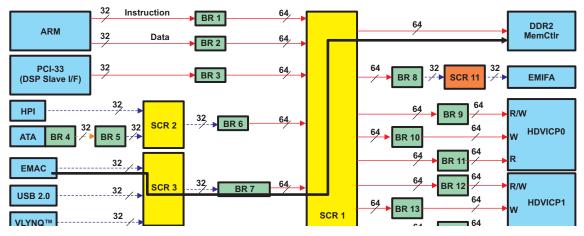


Figure 6. EMAC-to-DDR Transfer

This transaction experiences latencies in the master (EMAC), SCR3, Bridge7, SCR1, and the slave memory (DDR).



Also, it is important to note that accessing registers is not a single cycle access. It has to go through multiple SCRs/bridges and experiences hardware latency. Therefore, polling on registers costs more CPU cycles.

The latency faced in the bridges is directly related to the default burst size and the command FIFO depth. The worst latencies are due to SCR arbitration and bridge head of line blocking.

The topology is optimized to minimize latency between critical masters and slaves. For example, notice that in Figure 1 there are no bridges or extra infastructure elements (bridges, SCRs) from critical masters such as C64x+, and EDMA master to critical slaves such as C64x+ memory or DDR memory.

2.2 Head of Line Blocking

A command FIFO is implemented inside the bridge to queue transaction commands. All requests are queued on first-in-first-out basis; bridges do not reorder the commands. It is possible that a high priority request at the tail of a queue can be blocked by lower priority commands that could be at the head of the queue. This scenario is called bridge head of line blocking.

In Figure 7, the command FIFO size is 4. FIFO is completely filled with low priority (7) requests before a higher priority request (0) comes in. In this case, the high priority request has to wait until all four lower priority (7) requests get serviced.

When there are multiple masters vying for the same end point (or end points shared by the same bridge), the bridge head of line blocking is one of the factors that can affect system throughput and a master's ability to service read/write requests targeted to a slave peripheral/memory.

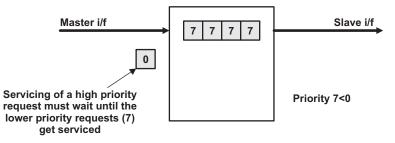


Figure 7. Bridge Head of Line Blocking

2.3 Reads Vs Writes

Note that read transactions are more costly than writes. In case of a read transaction, the master has to wait until it gets the data back from the slave. However, in a write case, typically the master can issue a write transaction and go ahead with the next transaction without waiting for a response from the slave.

On-chip memory access experiences less latency compared to off-chip. Off-chip memory is susceptible to extra latency (e.g., refresh cycles, CAS latency, etc.). If possible, it is recommended to keep frequently used code in on-chip memory for better system throughput performance.



2.4 Memory Maximum Bandwidths

Memory bandwidth has an effect on system throughput. More bandwidth gives better throughput performance. Table 4 shows all memories of the DM6467 SoC and their theoretical maximum bandwidths.

Theoretical maximum bandwidth is the maximum possible bandwith, which is calculated based on the memory clock and bandwith. This bandwith calculation assumes zero latency with SoC infrastructure elements.

Memory	Theoretical Maximum Bandwidth
ARM TCM	237.6 MB/s (ARM frequency * bus width / 5 ARM cycles per read/write operation = 297 MHz * 32-bit bus/5)
C6x+ DSP L1P/L2 RAM	2376 MB/s (C64x+ SDMA frequency * C64x+ SDMA bus width = 297 MHz * 64-bit bus)
DDR2	2376 MB/s (DDR2 clock frequency * DDR2 bus width = 297 MHz * 64-bit bus)
AEMIF	Depends on setup, strobe, and hold time configuration example: Read - $8.49MB/s$ with (setup/strobe/hold) =($6/26/3$) Write - $14.85MB/s$ with (setup/strobe/hold) =($6/11/3$)

Table 4. Memory Maximum Bandwidths

Refer to Table 10 for throughput numbers for EDMA transfers between these memories.

3 SoC Level Optimizations

This section describes techniques that can be implemented at the system level to optimize throughput. Section 4 describes optimation techniques that can be implemented at the IP or peripheral level.

3.1 SCR Arbitration

SCR provides priority-based arbitration to select the connection between master and slave peripherals; this arbitration is based on the priority value of each master.

Each master can have a priority value between 0 and 7 with 0 being the highest priority and 7 being the lowest priority. The prioritization scheme works such that, at any given time, if there are read/write requests from multiple masters vying for the same end point (same slave peripheral/memory or infrastructure component like bridge/SCR connecting to multiple slave peripherals), then the accesses from the master at the highest priority are selected first. Additionally, if there are read/write requests from masters programmed at the same/equal priority, then one request from each master is selected in a round-robin manner.

The prioritization within the SCR is programmable for each master by configuring the Bus Master Priority Control 0 Register (MSTPRI0), Bus Master Priority Control 1 Register (MSTPRI1), and Bus Master Priority Control 2 Register (MSTPRI2). For more details on these registers, see the *TMS320DM6467 Digital Media System-on-Chip Data Manual* (SPRS403). The default priority levels for the DM6467 SoC bus masters are shown in Table 5; lower values indicate higher priority.

Master	Default Priority Level
VPIF Capture	1
VPIF Display	1
TSIF0	1
TSIF1	1
EDMA3TC0	2
EDMA3TC1	2
EDMA3TC2	2
EDMA3TC3	2
HDVICP0 (CFG) ⁽¹⁾	3

Table 5. Default Master Priorities

(1) The C64x+ CFG, HDVICP0, and HDVICP1 priority values are not actually used by the SCR, which gives equal weight round-robin priority to accesses from these masters. The MSTRPRI register settings for these masters have no effect.

Master	Default Priority Level
HDVICP1 (CFG) ⁽¹⁾	3
ARM926 (INST)	4
ARM926 (DATA)	4
C64x+ (DMA)	4
C64x+ (CFG) ⁽¹⁾	4
VDCE	4
EMAC	5
USB2.0	5
ATA	5
VLYNQ	5
PCI	6
HPI	6

Table 5. Default Master Priorities (continued)

Note that there is no priority set for the EDMA3CC. This is because the EDMA3CC accesses only the TPTCs and is always given higher priority than the other masters on those Fast CFG SCR slave ports.

Although the default priority values (for different masters) have been chosen based on the prioritization requirements for the most common application scenarios, it is prudent to adjust/change the master priority values based on application-specific needs to obtain optimum system performance and to ensure real-time deadlines are met.

3.2 DDR2 Memory Controller Prioritization Scheme

The DDR2 memory controller services all master requests on priority basis and reorders requests to service highest priority requests first, improving system performance. For more details on the DDR2 memory controller prioritization scheme, see the *TMS320DM646x DMSoC DDR2 Memory Controller User's Guide* (SPRUEQ4).

3.3 C64x+ DSP Related Optimizations

3.3.1 Internal Direct Memory Access (IDMA)

The internal direct memory access (IDMA) controller is used to perform fast-block transfers between any two memories local to the C64x+ DSP. Local memories include Level 1 program (L1P), Level 1 data (L1D), and Level 2 (L2) memories. The IDMA is optimized for rapid burst transfers of memory blocks (contiguous data). The intent of the IDMA is to offload C64x+ DSP from on-chip memory (to/from L1D/L2) data movement tasks. For more details on the IDMA controller, see the *TMS320C64x*+ *DSP Megamodule Reference Guide* (SPRU871).

3.3.2 Choosing EDMA Vs CPU/IDMA

The following are a couple of points to keep in mind when choosing EDMA, CPU or IDMA for data transfers:

- The IDMA would give a better cycle/word performance than the EDMA for on-chip memory (to/from L1D/L2) transfers because IDMA is local to these memories, operates at a higher clock, and uses a bigger bus width.
- It is possible for certain on-chip memory (L1D/L2 to/from L2/L1D) transfer scenarios, both IDMA and CPU give nearly identical cycle/word efficiency. However, offloading the tasks of data transfers to IDMA allows more efficient usage of CPU bandwidth to perform other critical tasks.



www.ti.com



IP Throughput Optimization Techniques

www.ti.com

In summary, if concerned about L2 to L1 transfers, when geometry is fairly simple (i.e., 1-D xfer) and *performance* is the biggest care-about, then using IDMA makes the most sense. If you need extra flexibility and features (e.g., linking, chaining, 2-D xfer), then you can give up performance and use EDMA to perform these transfers. Note that competing accesses to these memories (by multiple masters) will impact the performance of IDMA.

4 IP Throughput Optimization Techniques

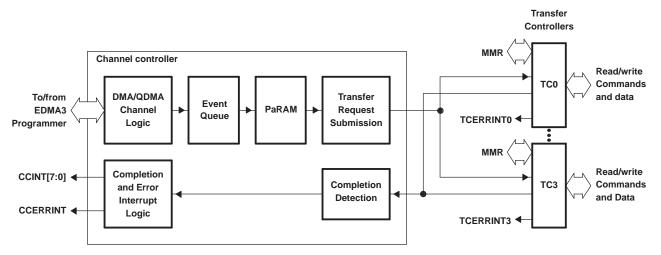
This section describes the throughput performance of different peripherals of the DM6467 SoC. It also provides the factors that affect peripheral throughput and recommendations for optimum peripheral performance.

4.1 Enhanced Direct Memory Access (EDMA)

This section provides a throughput analysis of the EDMA module integrated in the TMS320DM646x DMSoC.

4.1.1 Overview

The EDMA controller's primary purpose is to service user programmed data transfers between internal or external memory-mapped slave endpoints. It can also be configured for servicing event driven peripherals (such as serial ports), perform sorting or subframe extraction of various data structures, etc. There are 64 direct memory access (DMA) channels and 8 QDMA channels serviced by four concurrent physical channels. The block diagram of EDMA is shown in Figure 8.





DMA channels are triggered by external event, manual write to event set register (ESR), or chained event. QDMA are autotriggered when write is performed to the user-programmable trigger word.

Once a trigger event is recognized, the event is queued in the programmed event queue. If two events are detected simultaneously, then the lowest-numbered channel has highest priority.

Each event in the event queue is processed in the order it was queued. On reaching the head of the queue, the PaRAM associated with that event is read to determine the transfer details. The transfer request (TR) submission logic evaluates the validity of the TR and is submits a valid transfer request to the appropriate transfer controller. Figure 9 shows a block diagram of the channel controller.



From Peripherals/External Events

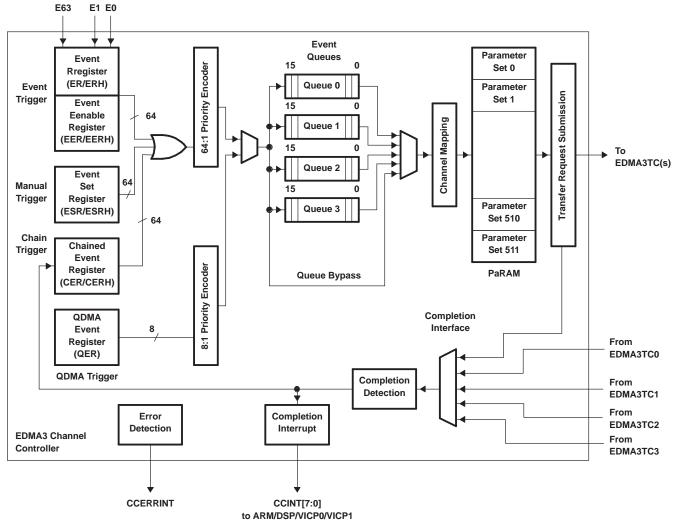


Figure 9. EDMA3 Channel Controller (EDMA3CC) Block Diagram



The transfer controller receives the request and is responsible for data movement as specified in the transfer request. Figure 10 shows a block diagram of the transfer controller.

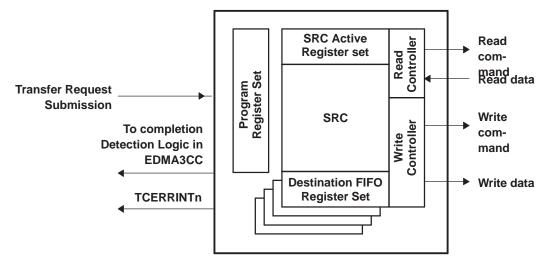


Figure 10. EDMA3 Transfer Controller (EDMA3TC) Block Diagram

The transfer controller receives the TR in the DMA program register set, where it transitions to the DMA source active set and the destination FIFO register set immediately. The read controller issues the read command when the data FIFO has space available for data read. When sufficient data is in the data FIFO, the write controller starts issuing the write command.

The maximum theoretical bandwidth for a given transfer can be found by multiplying the width of the interface and the frequency at which it transfers data. The maximum speed the transfer can achieve is equal to the bandwidth of the limiting port. In general, a given transfer scenario will never achieve maximum theoritical band width due to several factors, like transfer overheads, access latency of source/destination memories, finite number of cycles taken by EDMA3CC and EDMA3TC between the time the transfer event is registered to the time the first read command is issued to EDMA3TC. These overheads can be calibrated by looking at the time taken to do a 1 byte transfer. These factors are not excluded in these throughput measurements. Table 6 lists the internal bus frequencies at which different memories and slave end point operates and their bus widths.

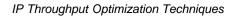
Module Name	Freq (MHz)	Bus Width (bits)
DDR2	297*2	32
AEMIF	Write: 14.85, Read: 8.49	16
HDVICP	297	64
GEM(L2/L1D/L1P)	297	64
ARM(TCM)	59.4	32

Table 6. Frequency and Bus Widths for Different Memory and Slave Endpoints	d Bus Widths for Different Memory and Slave Endpoints (1)	ths for Different Memory and Slave Endpoints ⁽¹⁾
--	---	---

⁽¹⁾ The CC/TC of EDMA3 is operating at divide-by 2 the CPU frequency.

The formulas used for the throughput calculations are shown below:

- Actual Throughput = (Transfer Size/Time Taken)
- Ideal Throughput = Frequency of Limiting Port * Data Bus Width in Bytes
- TC Utilization = (Actual Throughput/ Ideal Throughput) * 100





4.1.2 Test Environment

- The common system setup for the EDMA throughput measurement is given below::
- DSP clock: 594 MHz (unless specified)
- DDR clock: 297 MHz (unless specified)
- AEMIF configuration
 - Read time cycle (setup/strobe/hold): 35 (6/26/3)
 - Write time cycle (setup/strobe/hold): 20 (6/11/3)
 - Data bus width: 16 bits
- TCM memory: WAIT cycles enabled, takes 5 ARM cycles to access memory
- The data presented is for standalone transfers with no other ongoing or competing traffic
- All profiling done with CPU internal TSC timer

4.1.3 Factors Affecting EDMA Throughput Value

EDMA channel parameters allow many different transfer configurations. Typical transfer configurations result in transfer controllers bursting the read write data in default burst size chunks, thereby, keeping the busses fully utilized. However, in some configurations, the TC issues less than optimally sized read/write commands (less than default burst size), reducing performance. To properly design a system, it is important to know which configurations offer the best performance for high-speed operations. These considerations are especially important for memory to memory/paging transfers. Single-element transfer performance is latency-dominated and is unaffected by these conditions.

The different factors considered for throughput calculation with its impact is given in Table 7.

Factors	Impact	General Recommendation
Source/Destination Memory	The transfer speed depends on SRC/ DST memory bandwidth	Use L1, L2 or DDR for better results.
Transfer Size	Throughput is less for small transfers due to transfer overhead/latency	Configure EDMA for larger transfer size as throughput, small transfer size is dominated by transfer overhead
A-Sync/AB-Sync	Performance depends on the number of TRs. More TRs would mean more overhead.	
Source/Destination Bidx	Optimization will not be done if BIDX is not equal to ACNT value	Whenever possible, follow the EDMA3TC optimization guidelines
Queue TC Usage	Performance is the same for all four TCs	All four TCs have the same configuration and show the same performance
Burst Size	Decides the largest possible read/write command submission by TC	The default burst size for all transfer controllers is 32 bytes. This also results in most efficient transfers/throughput in most memory to memory transfer scenarios.
Source/Destination Alignment	Slight performance degradation if source/destination are not aligned to Default Burst Size (DBS) boundaries	For smaller transfers, as much as possible, source and destination addresses should be aligned across DBS boundaries
CPU and DDR Frequency	The utilization is above 90% when varying the CPU and the DDR frequency with respect to each other	

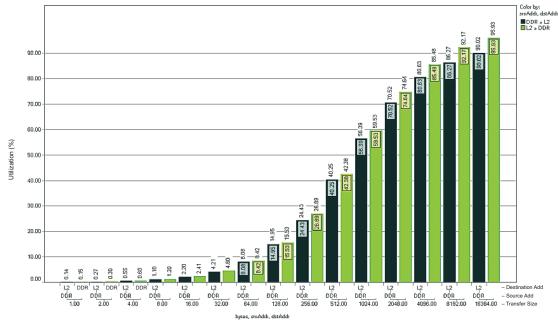
Table 7. Factors Considered for Throughput



IP Throughput Optimization Techniques

4.1.3.1 Transfer Size

Throughput is low for smaller transfer size (less than 1Kbytes) due to transfer overhead. Large transfer sizes give higher throughput. Figure 11 describes the percentage of utilization for transfers between L2 and DDR with different transfer sizes.



A For high-resolution image, see Figure A-1 in Appendix A.

Figure 11. Utilization of EDMA for L2, DDR Access

The Y-axis represents the percentage of utilization and X-axis represents the transfer size for DDR to L2 transfer. As transfer size increases, the impact of the transfer overhead/latency reduces which increases the percentage of utilization. In the analysis shown here, overhead includes the number of cycles from the logging of the start time stamp to event set, as well as the number of cycles from the EDMA completion interrupt to the logging of the end time stamp. The percentage of utilization is above 60% for a transfer with a data size greater than 1kB.

4.1.3.2 A-Sync/AB-Sync

An A-sync transfer is configured as follows:

- The number of TR submitted equals BCNT * CCNT
- Each sync event generates a TR with a transfer size equal to ACNT bytes

Therefore, this configuration results in the following trends:

• Larger ACNT values results in higher bus utilization by submitting larger transfer sizes per sync event which reduces transfer overhead.





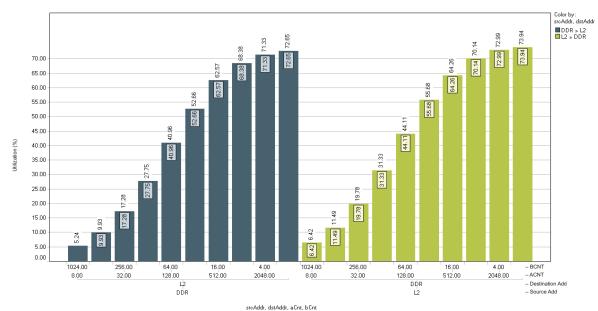


Figure 12 shows the percentage of utilization of EDMA for different ACNT value.

A For high-resolution image, see Figure A-2 in Appendix A.

Figure 12. Utilization for Different Element Size (ACNT)

The Y-axis represents the percentage of utilization and X-axis represents the different configuration of ACNT and BCNT value to do 8KB transfer between L2 and DDR memory locations. In the case of an AB-sync transfer, the number of TRs submitted is equal to CCNT; each TR causes the transfer of aCnt*bCnt bytes. If the number of TRs submitted for both A-sync and AB-sync is the same, then the throughput value will be almost the same.



IP Throughput Optimization Techniques

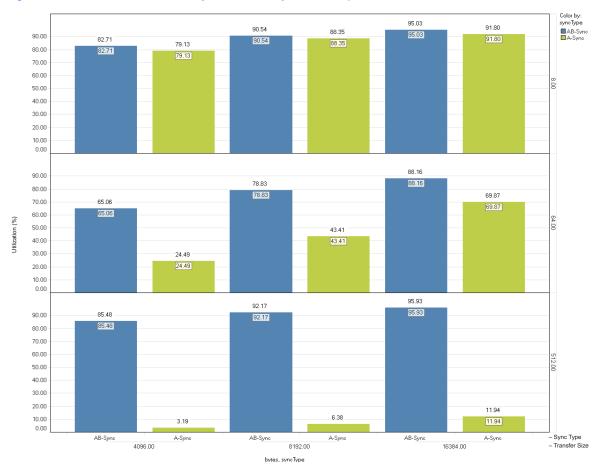


Figure 13 shows the effect of A-sync and AB-sync on the performance for different transfer size.

A For high-resolution image, see Figure A-3 in Appendix A.

Figure 13. Effect of A-Sync and AB-Sync

Figure 13 shows the comparison between A-Sync and AB-Sync transfers for different transfer size and varying BCNT (CCNT is always 1). The A-Sync transfers are done using chaining to self.

The y-axis represents the percentage of utilization for different BCNT (8/64/512); x-axis represents the A-sync and AB-sync for different transfer size. For BCNT equal to 8, the number of TRs submitted for A-sync is eight times more than the number of TRs submitted for AB-sync, which shows slight degradation in performance; whereas, for BCNT equal to 512, the number of TRs submitted for A-sync transfer is 512 times more than AB-sync, which shows huge degradation in performance.



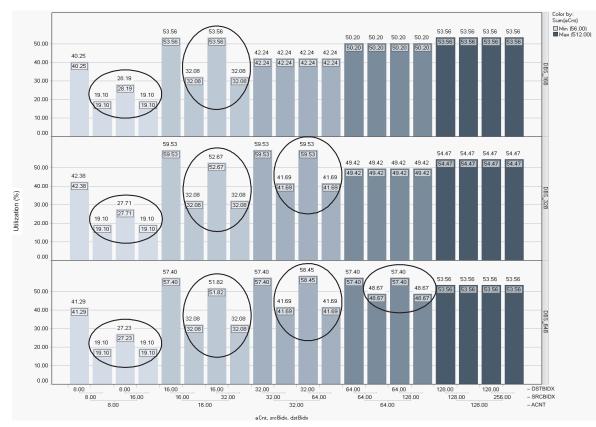
4.1.3.3 TC Optimization Rules

If ACNT <= DBS, default burst size, and ACNT is power of 2 and SRCDIDX/DSTBIDX = ACNT and BCNT <= 1023 and source address mode (SAM)/destination address mode (DAM) is Increment mode, the TC internally optimizes the transfer so that read and/or write commands treat the entire block transfer as a single linear transfer of ACNT = ACNT* BCNT rather than issuing just ACNT worth read/write commands. The read/write optimization rules are given in Table 8.

ACNT <= DBS	ACNT is Power of 2	BIDX = ACNT	BCNT<=1023	SAM/DAM Increment	Description
Yes	Yes	Yes	Yes	Yes	Optimized
No	Х	Х	Х	Х	Not Optimized
Х	No	Х	Х	Х	Not Optimized
Х	Х	No	Х	х	Not Optimized
Х	Х	Х	No	Х	Not Optimized

Table 8.	Read/Write	Command O	ptimization	Rules

Figure 14 shows the relative impact on performance for cases where both SRCBIDX/DSTBIDX = ACNT, SRCBIDX not equal to ACNT (TC only optimized the write commands), DSTBIDX not equal to ACNT (TC only optimize the read commands), and both SRCBIDX/DSTBIDX not equal to ACNT (in which case both read/write command optimization will not be performed by the TC). Similar degradation will be observed for cases where ACNT is not a power of 2 or BCNT is greater then 1023 or if SAM/DAM is not set to increment mode.



A For high-resolution image, see Figure A-4 in Appendix A.

Figure 14. Utilization for Different Destination Index Value



IP Throughput Optimization Techniques

www.ti.com

In Figure 14, Y-axis represents the percentage of utilization for different DBS and X-axis represents a combination of ACNT, srcBidx and dstBidx. This illustration is plotted for AB-Sync transfer mode. When the value of ACNT is less than the DBS, there is degradation in performance. The impact of non-linear indexing for ACNT smaller or equal to burst size is high lightened with circles on Figure 14.

For ACNT = 8, if SRCBIDX = 8 and DSTBIDX = 8 the utilization is better; however, for other combinations of BIDX it is low. This degradation is because the code optimization is not done for other combinations.

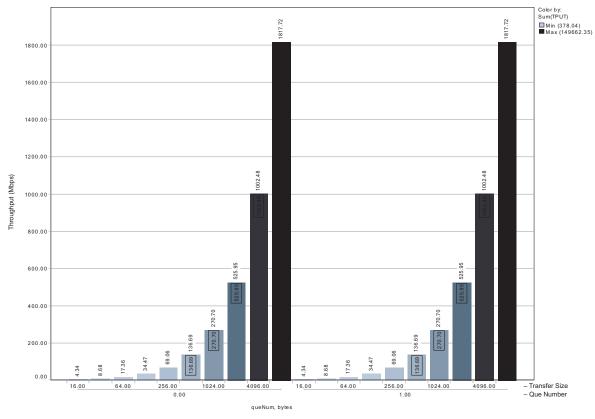
4.1.3.4 Queue TC Usage

On DM6467, there are four transfer controllers to move data between slave end points. The default configuration for the transfer controllers is shown in Table 9.

Name	TC0	TC1	TC2	TC3		
FIFOSIZE	256 bytes	256 bytes	256 bytes	256 bytes		
BUSWIDTH	8 bytes	8 bytes	8 bytes	8 bytes		
DSTREGDEPTH	4 entries	4 entries	4 entries	4 entries		
Default DBS	32 bytes	32 bytes	32 bytes	32 bytes		

Table 9. EDMA3 Transfer Controller Configurations

The individual TC performance for paging/memory to memory transfers is essentially dictated by the TC configuration. In most scenarios, the FIFOSIZE and default burst size configuration for the TC have the most significant impact on the TC performance; the BUSWIDTH configuration is dependent on the device architecture and the DSTREGDEPTH values impact the number of in flight transfers. On the DM6467 device, all transfer controllers yield identical performance for all transfer scenarios because all TC have the same configuration, and most importantly the same FIFOSIZE for a given burst size (which is configurable). Figure 15 shows the throughput of TC0 and TC1 for A-Sync(0) and AB-Sync(1).



A For high-resolution image, see Figure A-5 in Appendix A.

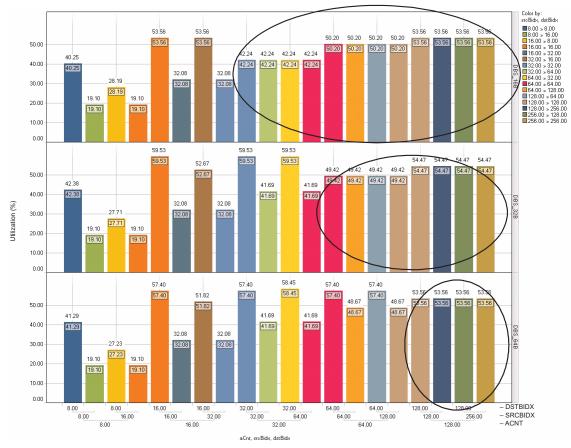




The Y-axis represents the throughput in MBps and the X-axis represents the different transfer size for both que 0 and que 1. These graphs show identical performance for TC0 and TC1. On the DM6467 device, all transfer controllers yield identical performance for all transfer scenarios because all TC have the same configuration, and most importantly the same FIFOSIZE for a given burst size, which is configurable.

4.1.3.5 Burst Size

The TC read and write controllers, in conjunction with the source and destination register sets, are responsible for issuing optimally-sized reads and writes to the slave endpoints. An optimally-sized command is defined by the transfer controller default burst size (DBS). Both of the read and write controller will always issue read commands that are always less-than-or-equal to DBS value. Figure 16 shows variation in the percentage of utilization with respect to different DBS value for transfer from DDR to L2.



A For high-resolution image, see Figure A-6 in Appendix A.



The Y-axis represents the percentage of utilization and the X-axis represents the configuration of SRCBIDX and DSTBIDX for different ACNT value. In Figure 16, ACNT is varied from 8 Bytes to 128 Bytes. The transfer controller attempts to issue the largest possible command size, as limited by the DBS value or ACNT/BCNT value for TR. The read/write controllers always issue the commands less than or equal to the DBS value. If the ACNT value is larger than the DBS value, then the TC breaks the ACNT array to in DBS-sized commands to the source and destination addresses; then, each BCNT number of arrays is serviced in succession as shown in the illustration by eclipse. If the ACNT value in AB-Sync transfer is less than or equal to the DBS value, then the TR may be optimized to in a 1-D transfer in order to maximize efficiency.

IP Throughput Optimization Techniques

www.ti.com

4.1.3.6 Source/Destination Alignment:

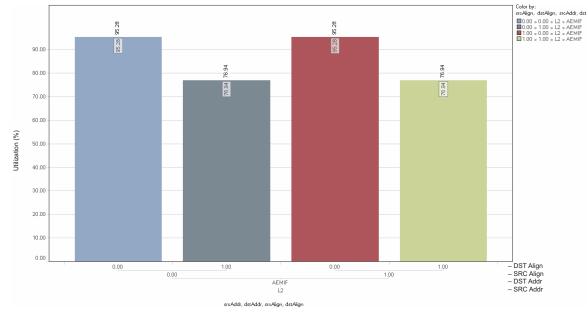


Figure 17 shows the effect of SRC/DST alignment for L2 to AEMIF transfer of 8KB of data.

A For high-resolution image, see Figure A-7 in Appendix A.

Figure 17. Utilization for Different Source and Destination Alignment

The Y-axis represents the percentage of utilization and the X-axis represents SRC alignment and DST alignment for L2 to AEMIF transfer. The performance is better if SRC and DST buffers are aligned at DBS boundaries.



4.1.3.7 CPU and DDR Frequency Variation

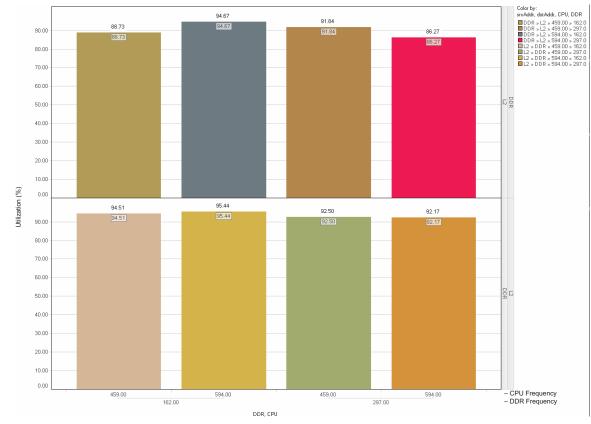


Figure 18 show the performance of EDMA for transfer between L2 and DDR for 16KB transfer size.

A For high-resolution image, see Figure A-8 in Appendix A.

Figure 18. Utilization for EDMA for Different CPU and DDR Frequency

The Y-axis represents the percentage of utilization for transfer between L2 and DDR and the X-axis represents different CPU frequency configuration for given DDR frequency. The utilization is above 85% for both combinations of CPU and DDR frequency.

4.1.4 Performance Of EDMA

Figure 19 and Table 10 capture the best case throughput and bus utilization for various source and destination memory combinations. Figure 19 shows % of utilization on Y-axis and various source and destination memory combinations on X-axis, color coded for extra clarity. Table 10 summarizes actual throughput and maximum typical throughput obtained in MBytes/sec along with % of utilization for different source and destination memory combinations. All data shown with ACNT equal to 8/16kB, BCNT and CCNT equal to 1, A-Sync transfers with increment addressing mode and CPU/DDR/memory setup as specified in Section 4.1.2.



IP Throughput Optimization Techniques

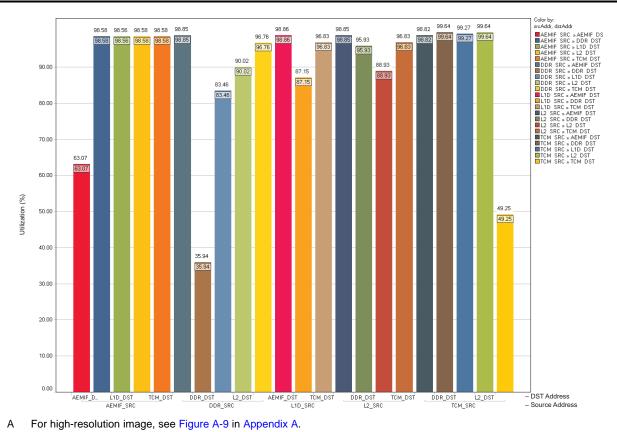


Figure 19. EDMA Performance

		Actual Throughput	Theoretical Maximum Throughput			
Source Mem	Destination Mem	(MBytes/sec)	(MBytes/sec)	Utilization (%)	Xfer Size (bytes)	
AEMIF_SRC	AEMIF_DST	5.35	8.49	63.07	16384	
	DDR_DST	8.37	8.49	98.58	16384	
	L1D_DST	8.36	8.49	98.56	8192	
	L2_DST	8.37	8.49	98.58	16384	
	TCM_DST	8.36	8.49	98.58	16384	
DDR_SRC	AEMIF_DST	14.68	14.85	98.85	16384	
	DDR_DST	853.84	2376	35.94	16384	
	L1D_DST	1982.9	2376	83.46	8192	
	L2_DST	2138.92	2376	90.02	16384	
	TCM_DST	229.9	237.6	96.76	8192	
L1D_SRC	AEMIF_DST	14.68	14.85	98.86	8192	
	DDR_DST	2070.66	2376	87.15	8192	
	TCM_DST	230.07	237.6	96.83	8192	

Table 10. Performan	ce of EDMA for 8KE	or 16KB Transfer
	CE OI EDIVIA IOI ORE	

www.ti.com



Source Mem	Destination Mem	Actual Throughput (MBytes/sec)	Theoretical Maximum Throughput (MBytes/sec)	Utilization (%)	Xfer Size (bytes)
L2_SRC	AEMIF_DST	14.68	14.85	98.85	8192
	DDR_DST	2279.18	2376	95.93	16384
	L2_DST	2112.92	2376	88.93	16384
	TCM_DST	230.07	237.6	96.83	8192
TCM_SRC	AEMIF_DST	14.67	14.85	98.82	16384
	DDR_DST	236.73	237.6	99.64	16384
	L1D_DST	235.87	237.6	99.27	8192
	L2_DST	236.73	237.6	99.64	16384
	TCM_DST	117.02	237.6	49.25	8192

Table 10. Performance of EDMA for 8KB or 16KB Transfer (continued)

4.2 Multichannel Audio Serial Port (McASP)

4.2.1 McASP Overview

The McASP functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. It is useful for time-division multiplexed (TDM) stream, inter-integrated sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT). The McASP consists of transmit and receive sections that can operate synchronized, or completely independently with separate master clocks, bit clocks, and frame syncs, and using different transmit modes with different bit-stream formats.

There are two instances of the McASP on this device: McASP0 and McASP1. The McASP0 module includes up to four serializers that can be individually enabled to either transmit or receive in all different modes. The McASP1 module is limited with only one pinned-out serializer that can only be enabled to transmit in DIT mode.

4.2.2 McASP Characterization

McASP is a slave peripheral that can be serviced by either the CPU or the EDMA. The CPU is mainly used to control the McASP register setup; the EDMA is mainly used to service the data required by the McASP. As shown in Figure 2, the audio CFG bus connecting to the McASP is 32-bit wide, and the McASP can be serviced through either its own DAT or the CFG port. The CFG port is mainly used for register configuration; the DAT port is mainly used for data transfer. The McASP data elements being serviced can be 8, 16, or 32 bit for each transfer. Even though the bus is 32-bit wide, only one data element is transferred during each clock cycle.

4.2.3 McASP Clocking

The McASP system clock is sourced from SYSCLK3, which is the PLL0 clock divided by 4. The McASP serial clock (clock at the bit rate) can be sourced from:

- Internally: passing through two clock dividers off the 24 MHz AUX_CLKIN clock
- Externally: directly from the ACLKR/ACLKX pin
- Mixed: an external clock is input to the McASP on either the AHCLKX or AHCLKR pin, and divided-down to produce the bit rate clock internally.

The McASP serial clock generators are able to produce two independent clock zones: transmit and receive. The serial clock generators can be programmed independently for the transmit section and the receive section, and may be completely asynchronous to each other. For more information on the clocking structure, see the *TMS320DM646x DMSoC Multichannel Audio Serial Port (McASP) User's Guide* (SPRUER1).



IP Throughput Optimization Techniques

www.ti.com

The McASP throughput is tightly related to the serial clock. In the current test environment, the McASP serial clock can only be sourced internally. Thus, the maximum serial clock rate is 24 MHz, obtained by setting the two clock dividers to 1. Therefore, for each serializer, the theoretical maximum throughput is 24 Mbps, regardless of receiving or transmitting (when the clock is 24 MHz). When all the serializers are activated (McASP0 has four serializers and McASP1 has only one serializer), the theoretical maximum throughput is 24 Mbps per serializer, regardless of receiving or transmitting.

4.2.4 Test Environment

The common system setup in this throughput analysis is as follows:

- DSP clock rate: 594 MHz
- DDR clock rate: 297 MHz
- AEMIF configuration
 - Read time cycle (setup/strobe/hold): 35 (6/26/3)
 - Write time cycle (setup/strobe/hold): 20 (6/11/3)
 - Data bus width: 16 bits
- McASP serial clock mode: sourced internally
- McASP master clock (AHCLKX/AHCLKR) rate: 24 MHz (set HCLKRDIV/HCLKXDIV to 1)
- · All four serializers are active during the analysis: two for transmitting and two for receiving
- This is a standalone McASP throughput analysis; the numbers might vary when additional peripherals are competing for system resources.

4.2.5 Factors Affecting McASP Throughput

Table 11 lists the factors that might affect McASP throughput.

Factor	Impact	General Recommendation
SRC/DST Buffer Location	Different memories have different EDMA access latencies. Too long an EDMA access might result in untimely service.	Avoid locating SCR/DST buffers in AEMIF memory due to its long access delay.
EDMA Queue Assignment	Assigning transmits and receive events to the same queue might add delay in servicing individual events, which might cause untimely service.	Still assign transmit and receive EDMA events to the same queue during general usage to save EDMA queue resource as the drawback is not significant.

Table 11. Factors Affecting McASP Throughput

Certain buffer locations and queue configurations might cause the EDMA to fail to service data in a timely manner. In such cases, too high a McASP bit clock rate leads to inaccurate data transfers or lost elements, eventually causing the McASP to malfunction. Therefore, the bit clock should be limited by the maximum rate that does not break the McASP operation, which results in sub-optimal throughput shown in the following cases. Experiment and analysis is done separately for 32-bit, 16-bity, and 8-bit elelment scenarios.



4.2.5.1 Case 1: 16- and 32-Bit Element Transfer

Figure 20 shows the throughput analysis for the 16- and 32-bit element.

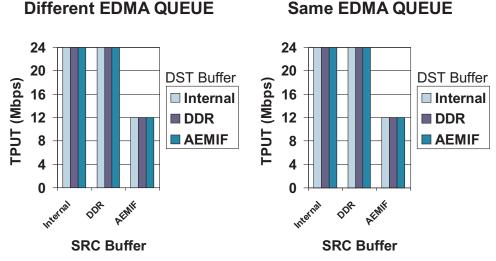


Figure 20. 16- and 32-Bit Element Throughput Analysis

4.2.5.1.1 SRC/DST Buffer Location

Internal memory has the shortest EDMA access latency for both read and write. Compared to that, DDR memory has a longer latency; AEMIF memory has the longest latency of all. For example, when SRC/DST buffers are both in internal memory or DDR memory, McASP can service data at the maximum bit clock, achieving 24 Mbps throughput. When the SRC buffer is in AEMIF memory, McASP can only maintain accurate transfer at 12 MHz bit clock regardless of the DST buffer location (set bit clock divisor to 2 – divisor value of 1 causes the McASP to transfer data incorrectly).

4.2.5.1.2 EDMA Queue Assignment

For 16- and 32-bit element data transfers, the same throughput can be obtained regardless of the EDMA queue assignment.

4.2.5.1.3 Optimization Recommendations

For 16- and 32-bit element data transfers, it is not recommended to set SRC/DST buffer in AEMIF memory due to its limited access speed. If space is sufficient, SRC/DST should be set in internal memory. If not, SRC/DST can be set in DDR memory provided that there is a small amount of traffic at the DDR bus. It is also recommended to assign the events to the same queue to minimize EDMA resource utilization. If the previous recommendations are followed, McASP can operate at any bit clock rate up to 24 MHz, which in turn produces a throughput equal to the theoretical number.

4.2.5.2 Case 2: 8-Bit Element Transfer

Figure 21 shows the throughput analysis for 8-bit element.

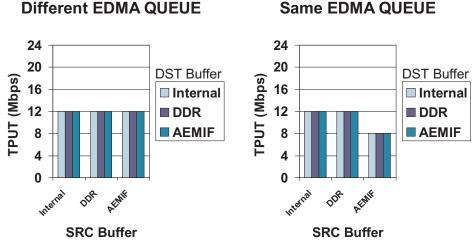


Figure 21. 8-Bit Element Throughput Analysis

4.2.5.2.1 SRC/DST Buffer Location

The same throughput of 12 Mbps can be obtained for all three memories even though they have different access latencies; divisor value of 1 (24 MHz bit clock) results in incorrect data transfers in the 8-bit transfer mode.

4.2.5.2.2 EDMA Queue Assignment

For 8-bit element data transfers, assigning receive and transmit EDMA events to the same EDMA queue results in a throughput of 8 Mbps when the SRC buffer is in AEMIF memory. Meanwhile, assigning the events to different queues can improve the throughput to 12 Mbps. When the SRC/DST buffers are in internal memories or DDR, a throughput of 12 Mbps can be obtained even when receive and transmit EDMA events are assigned to the same queue.

4.2.5.2.3 Optimization Recommendations

For 8-bit element data transfers, it is not recommended to set SRC/DST buffer in AEMIF memory due to its limited access speed. If space is sufficient, SRC/DST should be set in internal memory. If not, SRC/DST can be set in DDR memory provided that there is a small amount of traffic at the DDR bus. Moreover, it is recommended to assign EDMA events to the same queue to save EDMA resource unless SRC/DST buffers have to be set in AEMIF memory and 1.5M samples per second (per serializer) has to be maintained. If the previous recommendations are followed, McASP can operate at any bit clock rate up to 12 MHz, which in turn produces a throughput equal to 50% of the theoretical number.

4.3 Universal Asynchronous Receiver/Transmitter (UART)

4.3.1 UART Overview

The UART module in the TMS320DM646x DMSoC supports modem, infrared data (IrDA), and consumer infrared (CIR) functionalities. There are three UART instances in the DM646x devices: UART0, UART1, and UART2. For the detailed functionalities of the UART instances, see the *TMS320DM646x DMSoC Universal Asynchronous Receiver/Transmitter (UART) User's Guide* (<u>SPRUER6</u>).



The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link. This module is also capable of performing standard infrared communication in slow infrared mode (SIR) and medium infrared mode (MIR) defined by the Infrared Data Association. This module also supports consumer infrared (CIR) communications. The CIR mode uses a variable pulse width modulation technique to encompass the various formats of infrared encoding for remote control applications. The CIR logic is to transmit and receive data packets according to the user-definable frame structure and packet content.

4.3.2 UART Characterization

UART is a slave peripheral that can be serviced by either the CPU or EDMA. The CPU is mainly used to control the UART register setup; the EDMA is mainly used to service the data required by the UART. The system bus connecting to the UART is 8-bit wide; the UART element size is up to 8-bit. The UART RX and TX FIFO, which can be enabled or disabled, are both 64-byte deep.

4.3.3 UART Clocking

The UART system clock is sourced from SYSCLK3, which is the PLL0 clock divided by 4. The UART serial clock is source from the 24 MHz AUX_CLKIN clock.

The UART throughput is tightly related to the 24 MHz serial clock. Therefore, the theoretical maximum throughput is 1.8462 Mbps regardless of receiving or transmitting in modem mode. In the case of SIR communication, UART can receive/transmit up to 57.692 Kbps; in the case of MIR communication, UART can receive/transmit at 0.5736 Mbps. CIR bit rate is generally much slower than that of the modem mode, and is not considered in the throughput calculation. Therefore, modem mode will be the only mode considered in this application note. Table 12 shows the detailed modem mode baud rate. Table 13 shows the detailed IrDA mode baud rate.

Baud Rate	Baud Multiple	DLH,DLL (Decimal)	DLH,DLL (Hex)	Actual Baud Rate	Error (%)
0.3 Kb/s	16x	5000	13h, 88h	0.3 Kb/s	0
1.2 Kb/s	16x	1250	4h, E2h	1.2 Kb/s	0
2.4 Kb/s	16x	625	2h, 71h	2.4 Kb/s	0
14.4 Kb/s	16x	104	0, 68h	14.423 Kb/s	+0.16
28.8 Kb/s	16x	52	0, 34h	28.864 Kb/s	+0.16
57.6 Kb/s	16x	26	0, 1Ah	57.692 Kb/s	+0.16
115.2 Kb/s	16x	13	0, Dh	115.38 Kb/s	+0.16
230.4 Kb/s	13x	8	0, 8h	230.77 Kb/s	+0.16
460.8 Kb/s	13x	4	0. 4h	461.54 Kb/s	+0.16
921.6 Kb/s	13x	2	0, 2h	923.08 Mb/s	+0.16
1.8432 Kb/s	13x	1	0, 1h	1.8462Mb/s	+0.16

Table 12. UART Modem Mode Baud Rate

Table 13. UART IrDA Mode Baud Rate

Baud Rate	IR Mode	Baud Multiple	Encoding	DLH,DLL	Actual Baud Rate (*=Avg)	Error (%)
2.4	SIR	16x	3/16	625	2.4 kb/s	0
9.6	SIR	16x	3/16	156	9.6153 Kb/s	+0.16
19.2	SIR	16x	3/16	78	19.231 Kb/s	+0.16
57.6	SIR	16x	3/16	26	57.692 Kb/s	+0.16
0.576	MIR	41x/42x	1/4	1	0.5756 Mb/s	0



4.3.4 Test Environment

The common system setup in this throughput analysis is as follows:

- DSP clock rate: 594 MHz
- DDR clock rate: 297 MHz
- AEMIF configuration
 - Read time cycle (setup/strobe/hold): 35 (6/26/3)
 - Write time cycle (setup/strobe/hold): 20 (6/11/3)
 - Data bus width: 16 bits
- UART serial clock: 24 MHz
- UART RX FIFO trigger level to halt transmission: set to a minimum (0x0)
- UART RX FIFO trigger level to restore transmission: set to a maximum (0xF)
- UART transmitter and receiver are activated at the same time
- This is a standalone UART throughput analysis (the numbers might vary when additional peripherals are competing for system resources)

4.3.5 UART Throughput Information

Table 14 lists the factors that might affect UART throughput.

Factor	Impact
SRC/DST Buffer Location	Different memories have different EDMA access latencies. Too long an EDMA access might cause UART FIFO underflow/overflow.
TX/RX FIFO Trigger Level	Low receive FIFO or high transmit FIFO level implies more EDMA events, which will require EDMA to service UART more frequently with less number of bytes per event. This EDMA access latecy might result in untimely service.

Table 14. Possible Effective Factors of UART Throughput

In this analysis, the EDMA setup for servicing the DM6467 UART strictly followed the setup guideline in FIFO DMA Mode Operation section of the *TMS320DM646x DMSoC Universal Asynchronous Receiver/Transmitter (UART) User's Guide* (<u>SPRUER6</u>). For detailed EDMA parameter setup guidelines for servicing the DM6467 UART, see the *TMS320DM646x DMSoC Enhanced Direct Memory Access (EDMA3) User's Guide* (<u>SPRUEQ5</u>) and the *TMS320DM646x DMSoC Universal Asynchronous Receiver/Transmitter (UART) User's Guide* (<u>SPRUEQ5</u>).



4.3.5.1 SRC/DST Buffer Location

A snapshot of the analysis performed on the dependency that the throughput has on the source and destination buffer location is provided in Figure 22.

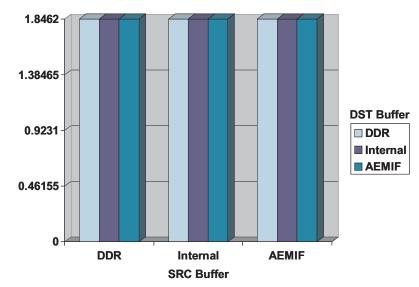
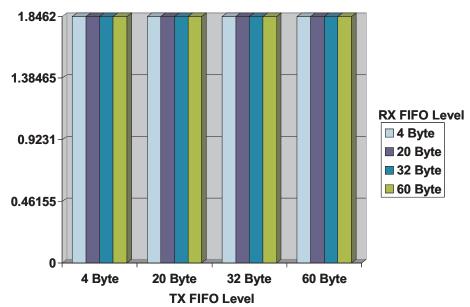


Figure 22. Throughput Dependency on SRC/DST Buffer Location (TX Queue = 1, RX Queue = 1, TX Trigger Level = 20, RX Trigger Level = 60)

The experiment result shows that UART throughput is not dependent on the SRC/DST buffer location in our standalone analysis. The same performance can be achieved for any combination of SRC/DST buffers.

4.3.5.2 TX/RX FIFO Trigger Level

A snapshot of the analysis performed on the dependency that the throughput has on the TX and RX FIFO trigger level is provided in Figure 23.







IP Throughput Optimization Techniques

www.ti.com

The experiment result shows that the UART throughput is not dependent on the TX/RX FIFO trigger level in this standalone analysis. The same performance can be achieved for any combination of the TX and RX FIFO trigger level.

4.3.5.3 Conclusion

In our standalone analysis, when the UART and EDMA setup listed in the UART and EDMA PRG is followed, UART can accurately operate at any rate included in Figure 22. The performance of UART is not affected by any of the following factors:

- SRC/DST buffer location or alignment
- TX FIFO/RX FIFO trigger level
 - **Note:** Even though unaffected in the standalone analysis, the throughput might change if heavy traffic on the memory bus is present or EDMA resource is limited.

4.4 Ethernet Media Access Controller (EMAC)

This section provides a throughput analysis of the EMAC module integrated in the TMS320DM646x DMSoC.

4.4.1 Overview

The EMAC module is used to move the data between the DM646x DMSoC and another host connected to the same network, in compliance with the Ethernet protocol.

Figure 24 shows the three main functional modules of the EMAC/management data input/output (MDIO) peripheral:

- EMAC control module
- EMAC module
- MDIO module

The EMAC control module is the main interface between the device core processor, EMAC module, and MDIO module. The EMAC control module incorporates 8K-bytes internal RAM to hold the EMAC buffer descriptors.

The MDIO module implements the 802.3 serial management interface to interrogate and control up to 32 Ethernet PHYs connected to the device, using shared two-wire bus. The host software uses the MDIO module to configure the autonegotiation parameters of each PHY attached to the EMAC, retrieve the negotiation result, and configure required parameters in the EMAC module for correct operation. The module is designed to allow almost transparent operation of the MDIO interface, with very little maintenance from the processor.

The EMAC module provides an efficient interface between the processor and the networked community. The EMAC on the device supports 10BaseT (10Mbits/second) and 100BaseT (100Mbits/second) in either half-duplex or full-duplex mode and 1000BaseT (1000Mbits/second) in full duplex mode, with hardware flow control and quality-of service (QOS) support.



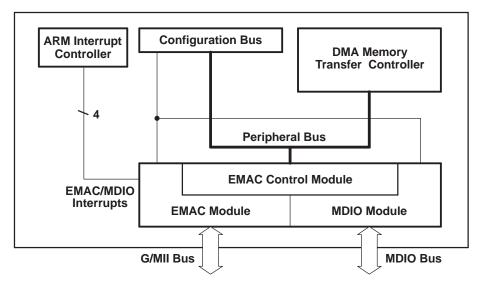


Figure 24. EMAC and MDIO Block Diagram

The EMAC control module can access both internal and external memory through the DMA memory transfer controller. The configuration bus is used to configure the control registers of the EMAC control module, EMAC module, and MDIO module. The EMAC and MDIO interrupts are combined into the EMAC control module which goes to the interrupt controller.

The format of an Ethernet frame is shown in Figure 25 and described in Table 15. The data portion of a single Ethernet frame on the wire is shown outlined in bold. The Ethernet frames are of variable lengths, with no frame smaller than 64 bytes or larger than RXMAXLEN bytes.

Number of Bytes

7	1	6	6	2	46–1500	4
Preamble	SFD	Destination	Source	Len	Data	FCS

Legend: SFD=Start Frame Delimeter; FCS=Frame Check Sequence (CRC)

Figure 25. Ethernet Frame Format



Field	Bytes	Description
Preamble	7	Preamble. These 7 bytes have a fixed value of 55h and serve to wake up the receiving EMAC ports and to synchronize their clocks to that of the sender's clock.
SFD	1	Start of Frame Delimiter. This field with a value of 5Dh immediately follows the preamble pattern and indicates the start of important data.
Destination	6	Destination Address. This field contains the Ethernet MAC address of the EMAC port for which the frame is intended. It may be an individual or multicast (including broadcast) address. When the destination EMAC port receives an Ethernet frame with a destination address that does not match any of its MAC physical addresses, and no promiscuous, multicast or broadcast channel is enabled, it discards the frame.
Source	6	Source Address. This field contains the MAC address of the Ethernet port that transmits the frame to the Local Area Network.
Len	2	Length/Type field. The length field indicates the number of EMAC client data bytes contained in the subsequent data field of the frame. This field can also be used to identify the type of data the frame is carrying.
Data	46 to (RXMAXLEN - 18)	Data Field. This field carries the datagram containing the upper layer protocol frame, that is, IP layer datagram. The maximum transfer unit (MTU) of Ethernet is (RXMAXLEN - 18) bytes. This means that if the upper layer protocol datagram exceeds (RXMAXLEN - 18) bytes, then the host has to fragment the datagram and send it in multiple Ethernet packets. The minimum size of the data field is 46 bytes. This means that if the upper layer datagram is less than 46 bytes, the data field has to be extended to 46 bytes by appending extra bits after the data field, but prior to calculating and appending the FCS.
FCS	4	Frame Check Sequence. A cyclic redundancy check (CRC) is used by the transmit and receive algorithms to generate a CRC value for the FCS field. The frame check sequence covers the 60 to (RXMAXLEN - 4) bytes of the packet data. Note that this 4-byte field may or may not be included as part of the packet data, depending on how the EMAC is configured.

Table 15. Ethernet Frame Description

4.4.2 Test Environment

The common system setup for the EMAC throughput measurement is given below:

- All tests are performed on DM6467 revision 1.1
- CPU clock frequency is 594 MHz
- DDR clock frequency is 297 MHz
- AEMIF clock configuration is:
 - Read time cycles (setup/strobe/hold): 35 (6/26/3)
 - Write time cycles (setup/strobe/hold): 20 (6/11/3)
 - Data bus width
- TCM memory: WAIT cycles enabled, takes 5 ARM cycles to access memory
- Throughput data collected is standalone, no other ongoing traffic
- Readings are collected for 100 Mbps and 1000 Mbps



4.4.3 Factors Affecting the EMAC Throughput

The EMAC throughput depends on the packet size, the type of memory it accesses through DMA and the descriptors memory location. To properly configure a system, it is important to know which configuration offers the best performance. The different factors and their impact on the EMAC throughput are given in Table 16.

Factor	Impact	General Recommendation	
Packet Size Performance is less for small packet size due to Configure EMAC for transfer overhead/latency		Configure EMAC for large packet size.	
Descriptor Memory Location	Performance is less for slow memory like AEMIF	Configure descriptor memory location to the EMAC internal RAM for best performance.	
Source Memory Location	Performance is less for slow memory like AEMIF	Configure source memory location to the GEM internal memory (e.g., L2)	
Destination Memory Location	Performance is less for slow memory like AEMIF	Configure destination memory location to the DDR, TCM or GEM internal memory (e.g., L2)	

Table 16. Factors Considered for Throughput

4.4.3.1 Packet Size

Figure 26 shows the effect of the different packet sizes on the EMAC throughput for both 100 Mbps and 1000 Mbps modes. The number of packets transferred is 10. As the packet size increases, the throughput value also increases irrespective of source address, destination address and descriptor address. Once the packet is transferred, the EMAC or CPU accesses the next descriptor address until the end of all the packets. The descriptor memory is accessed more frequently if the packet size is less than the EMAC or CPU, which adds more delay in total transfer time and causes degradation in the throughput for smaller packet size. Figure 26 shows that the EMAC performance is better if the packet size is above 400Bytes.

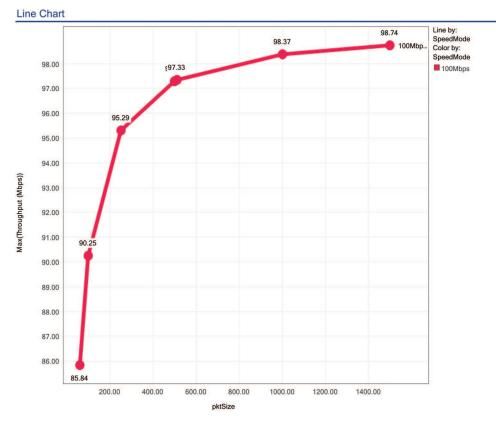


Figure 26. Effect of Packet Size on the EMAC Throughput for 100 Mbps Mode

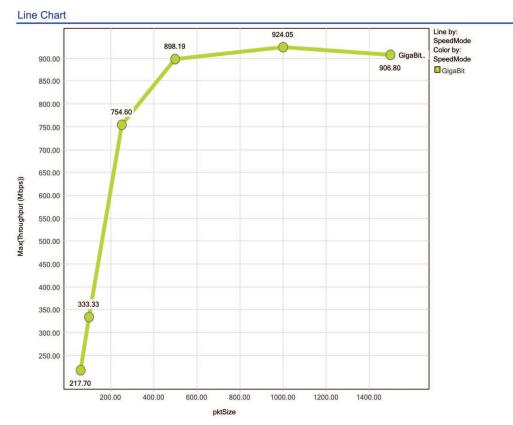


Figure 27. Effect of Packet Size on the EMAC Throughput for Giga Bit Mode



4.4.3.2 Descriptor Memory Location

Figure 28 shows the effect of the descriptor memory location on the EMAC throughput. The EMAC or CPU accesses the descriptor memory for each transfer. It takes more time if it is placed in slow memory, which causes degradation in the performance. For better performance, the descriptor memory should be placed in fast memory (i.e., L2 and DDR). EMAC has internal memory of 8KB to hold the descriptors and can be configured to transfer up to 512 packets without any CPU intervention. Figure 28 shows that the performance is better if the descriptors are kept in the EMAC internal RAM.

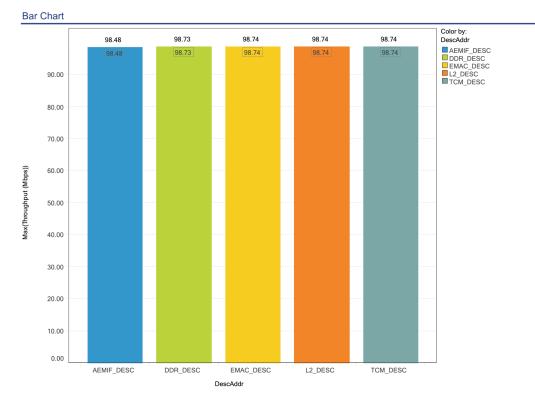


Figure 28. Effect of Descriptor Memory Location on the EMAC Throughput for 100 Mbps mode



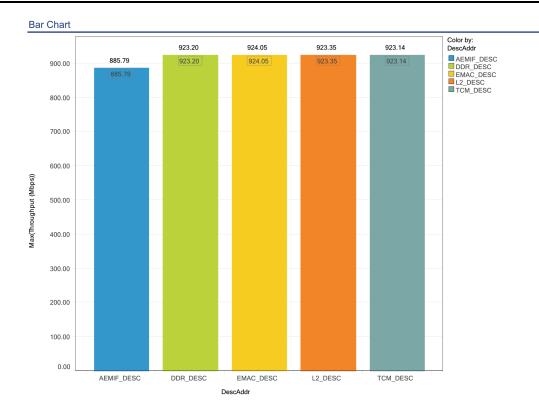


Figure 29. Effect of Descriptor Memory Location on the EMAC Throughput for Giga Bit Mode





4.4.3.3 Source Memory Location

Figure 30 shows the effect of the source memory location on the EMAC throughput for 100 Mbps and 1000 Mbps. Once the EMAC transfer is triggered, the DMA accesses the data from the source memory address. DMA takes more time to access If the source data is kept in the slow memory (i.e., AEMIF), which causes degradation in the performance. For better performance, source memory must be kept in fast memory (i.e., L2 or DDR).

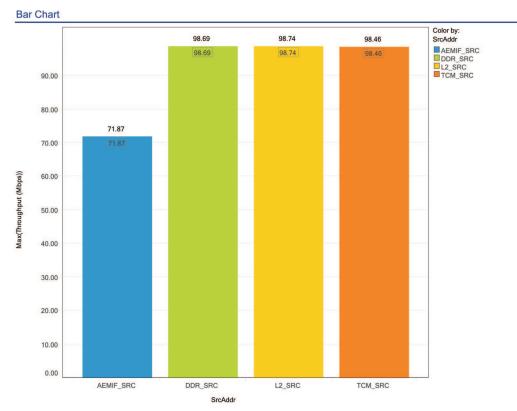


Figure 30. Effect of Source Memory Location on the EMAC Throughput for 100 Mbps Mode



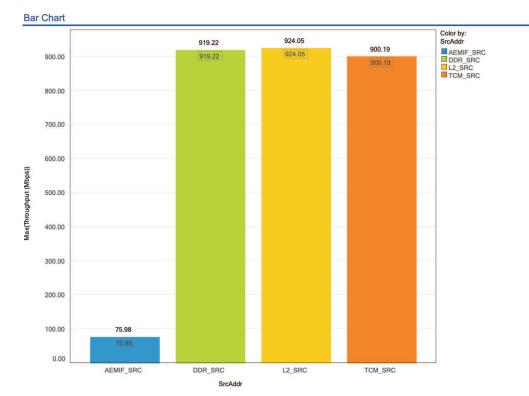


Figure 31. Effect of Source Memory Location on the EMAC Throughput for Giga Bit mode



4.4.3.4 Destination Memory Location

Figure 32 shows the effect of the destination memory location on the EMAC throughput for 100 Mbps and 1000 Mbps. Once the EMAC transfer is triggered, DMA puts the data to the destination memory address. It takes more time to access DMA if the destination memory is configured to slow memory (i.e., AEMIF), which causes degradation in the performance. For better performance, destination memory must be kept in fast memory like L2 or DDR.

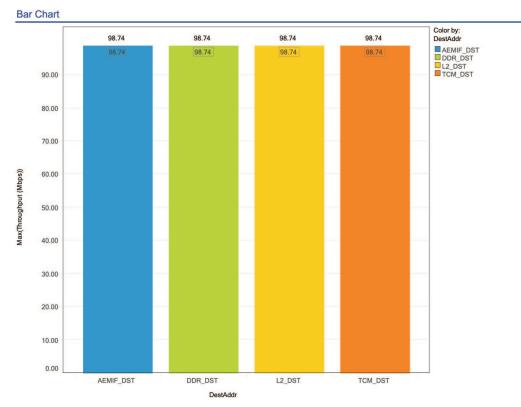


Figure 32. Effect of Destination Memory Location on the EMAC Throughput for 100 Mbps Mode

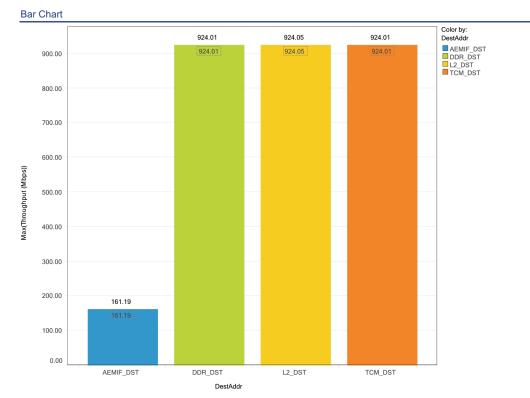


Figure 33. Effect of Destination Memory Location on the EMAC Throughput for Giga Bit Mode



4.4.4 The Best EMAC Configuration

Figure 34 shows the EMAC throughput value for all combinations of descriptor memory locations, source memory locations and destination memory locations. The packet size is configured for 1500 bytes and the packet number is configured for 10 packets. From Figure 34, the best configuration for EMAC is given below:

- Descriptor memory location: EMAC internal RAM
- Source memory location: CPU internal memory (e.g., L2)
- Destination memory location: DDR, TCM and L2

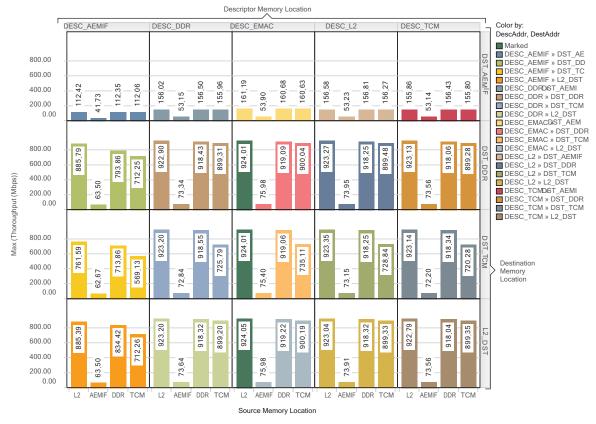


Figure 34. Effect of Different Memory Locations on the EMAC Throughput

Texas Instruments

References

www.ti.com

Table 17 shows the throughput values for all combinations of descriptor memory locations, source memory locations and destination memory. The best EMAC configuration is shown in bold in Table 17.

		DESC_AEMIF	DESC_DDR	DESC_EMAC	DESC_L2	DESC_TCM
L2_SRC	DST AEMIF	112.42	156.02	161.19	156.58	155.86
	DST_DDR	885.79	922.90	924.01	923.27	923.13
	DST_TCM	761.59	923.20	924.01	923.35	923.14
	L2_DST	885.39	923.20	924.05	923.04	922.79
SRC_AEMIF	DST_AEMIF	41.73	53.15	53.90	53.23	53.14
	DST_DDR	63.50	73.34	75.98	73.95	73.56
	DST_TCM	62.67	72.84	75.40	73.15	72.20
	L2_DST	63.50	73.64	75.98	73.91	73.56
SRC_DDR	DST_AEMIF	112.35	156.50	160.68	156.81	156.43
	DST_DDR	793.86	918.43	919.09	918.25	918.06
	DST_TCM	713.86	918.55	919.06	918.25	918.34
	L2_DST	834.42	918.32	919.22	918.32	918.04
SRC_TCM	DST_AEMIF	112.06	155.96	160.63	156.27	155.80
	DST_DDR	712.25	899.31	900.04	899.48	899.28
	DST_TCM	569.13	725.79	735.11	728.84	720.28
	L2_DST	712.26	899.20	900.19	899.33	899.35

Table 17. Effect of Different Memory on the EMAC Throughput

5 References

- TMS320DM6467 Digital Media System-on-Chip Data Manual (SPRS403)
- TMS320DM646x DMSoC Universal Asynchronous Receiver/Transmitter (UART) User's Guide (<u>SPRUER6</u>)
- TMS320DM646x DMSoC Enhanced Direct Memory Access (EDMA3) User's Guide (SPRUEQ5)
- TMS320DM646x DMSoC Multichannel Audio Serial Port (McASP) User's Guide (SPRUER1)
- TMS320DM646x DMSoC DDR2 Memory Controller User's Guide (SPRUEQ4)
- TMS320C64x+ DSP Megamodule Reference Guide (SPRU871)



Appendix A EDMA High-Resolution Diagrams

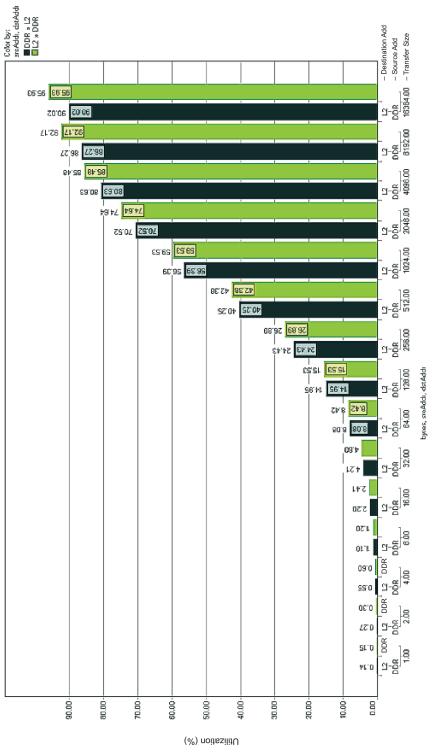


Figure A-1. Utiliztation of EDMA for L2, DDR Access



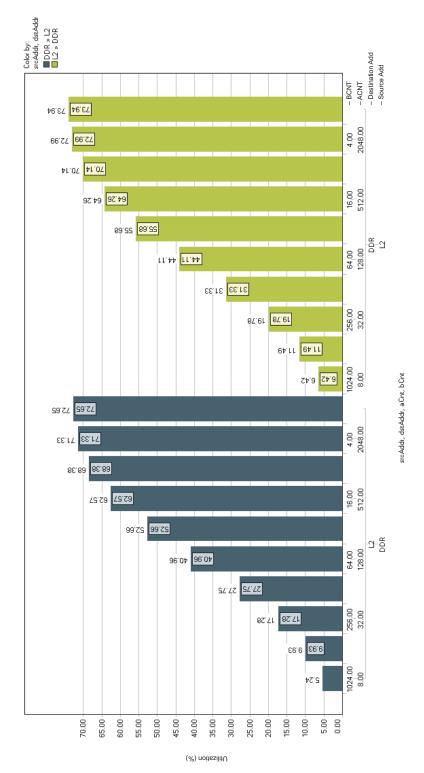


Figure A-2. Utilization for Different Element Size (ACNT)



Texas Instruments

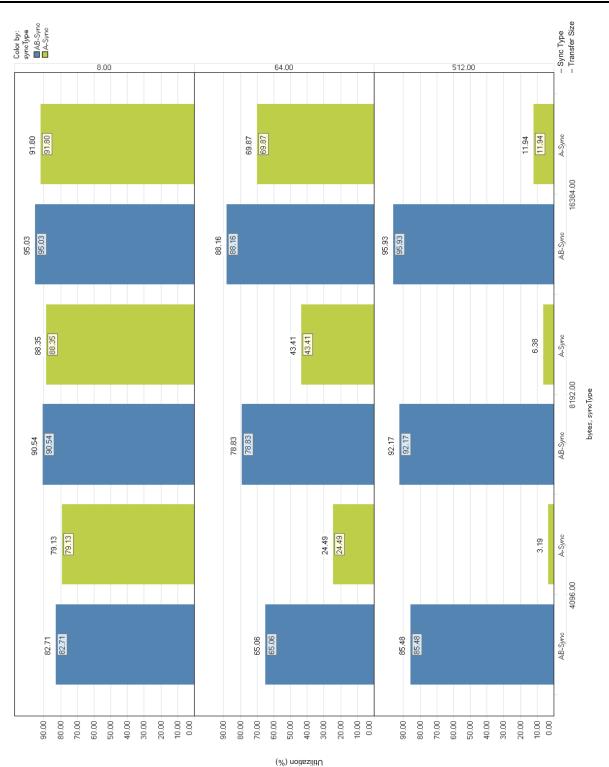


Figure A-3. Effect of A-Sync and AB-Sync



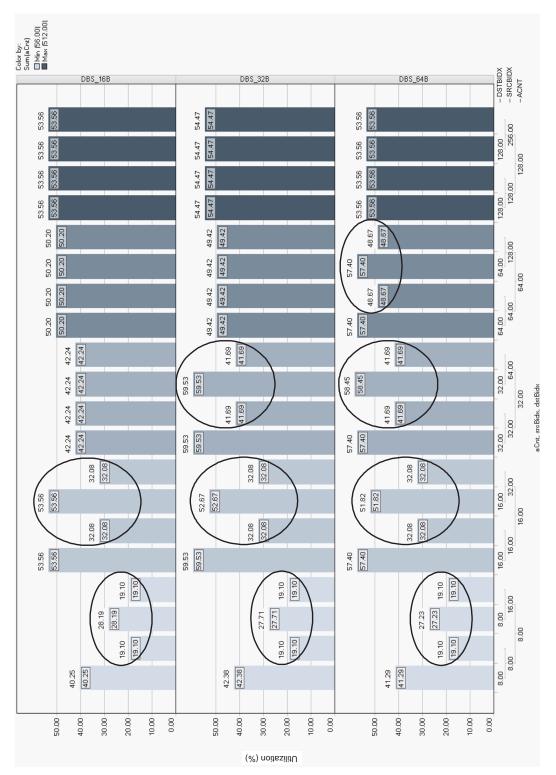


Figure A-4. Utilization for Different Destination Index Value



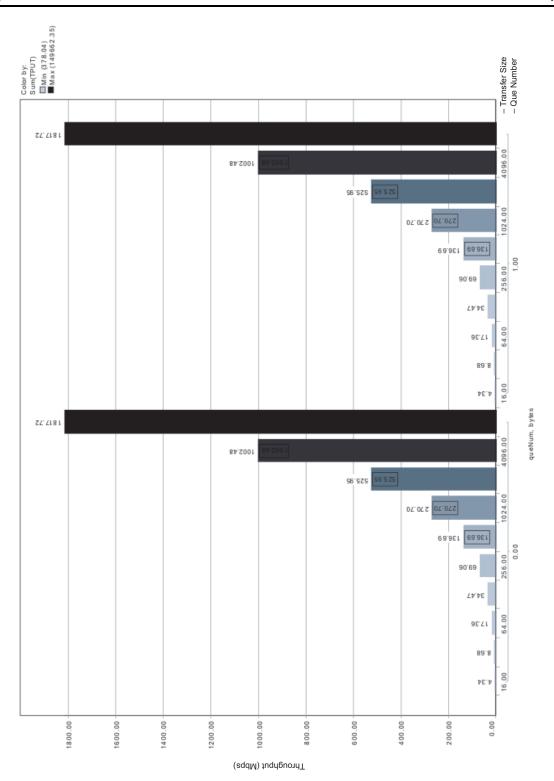


Figure A-5. Performance of TC0 and TC1

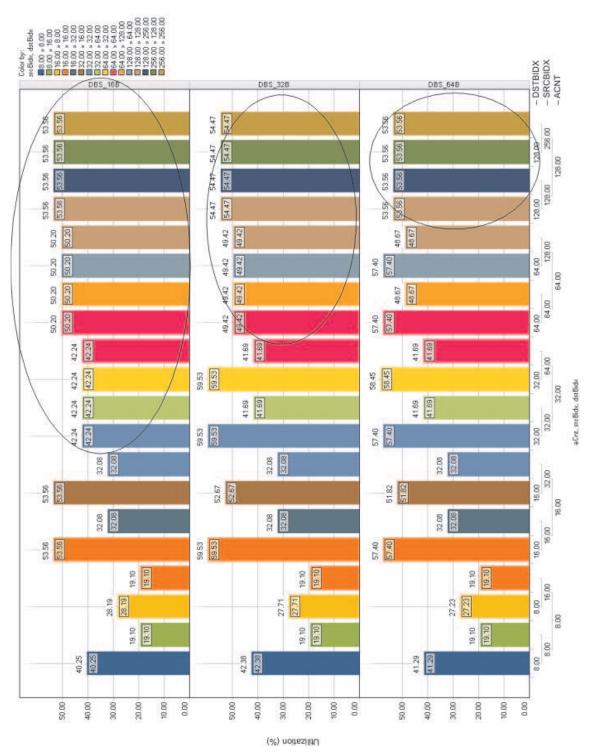


Figure A-6. Utilization for Different Burst Size Configuration





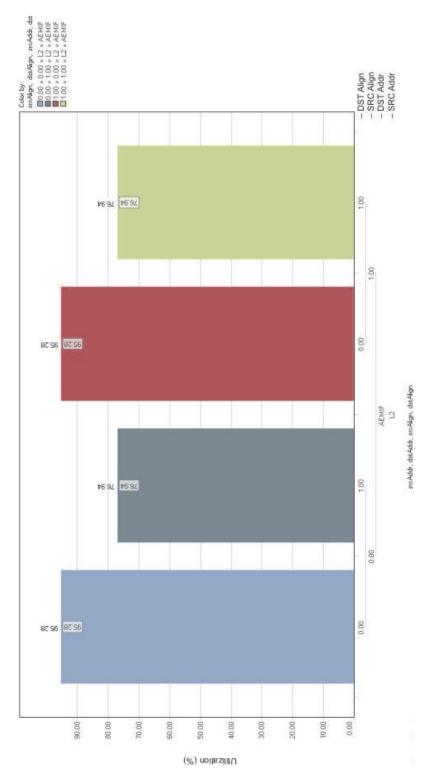


Figure A-7. Utilization for Different Source and Destination Alignment

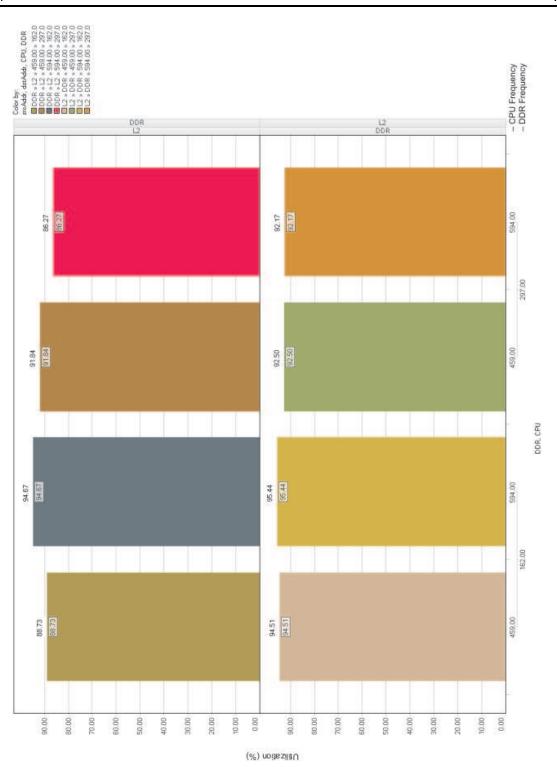


Figure A-8. Utilization for EDMA for Different CPU and DDR Frequency





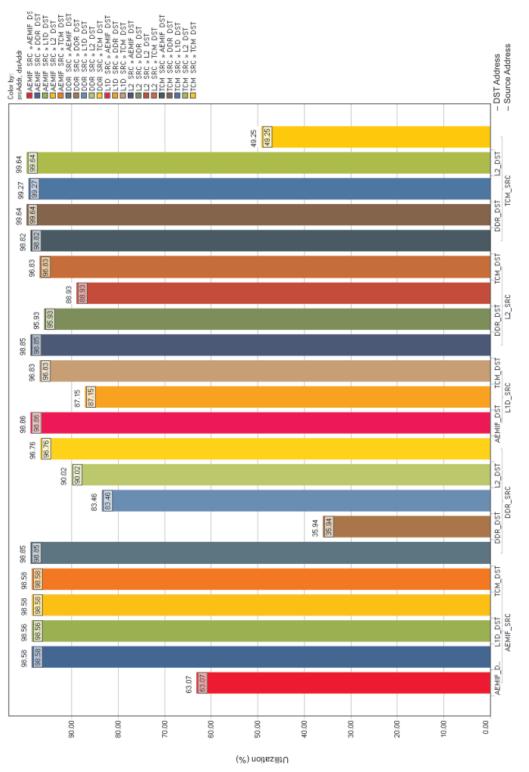


Figure A-9. EDMA Performance

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		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

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