

# **TMS320DM648/7 SoC Architecture and Throughput Overview**

Manoj Bohra

## **ABSTRACT**

This application report provides information on the DM648/7 throughput performance and describes the DM648/7 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 .....	8
3	3 SoC Level Optimizations.....	10
4	IP Throughput .....	11
5	References.....	51

## **List of Figures**

1	TMS320DM648/7 System Interconnect Block Diagram .....	3
2	TMS320DM648/7 Peripheral Configuration Bus.....	4
3	Bridge .....	6
4	Bus-Width and Clock Rate Conversion .....	6
5	Bridge Head of Line Blocking .....	7
6	PCI-to-DDR Transfer .....	9
7	EDMA3 Controller Block Diagram .....	12
8	EDMA3 Channel Controller (EDMA3CC) Block Diagram .....	13
9	EDMA3 Transfer Controller (EDMA3TC) Block Diagram .....	14
10	Utilization of EDMA for L2, DDR Access.....	16
11	Throughput of EDMA for L2, DDR Access .....	17
12	Utilization for Different Element Size (ACNT) .....	18
13	Effect of A-Sync and AB-Sync.....	18
14	Utilization for Different Destination Index Value.....	20
15	Performance of TC0 and TC1 .....	21
16	Performance of TC0 and TC1 .....	22
17	Utilization for TC0 and TC2 for Different CPU and DDR Frequency .....	23
18	Max throughput for TC0 and TC2 for Different CPU and DDR Frequency .....	23
19	EDMA Performance.....	25
20	EDMA Performance.....	25
21	16- and 32-Bit Element Throughput Analysis .....	28
22	Video Port Functional Block Diagram .....	29
23	Throughput for 8 Captures in 8-bit BT.656 Capture Mode .....	33
24	Utilization for 8 Captures in 8-bit BT.656 Capture Mode .....	34
25	Throughput for Single Channel 8-Bit BT.656 Display Mode .....	35
26	Utilization for Single Channel 8-Bit BT.656 Display Mode .....	36
27	Throughput for Single Channel 8-Bit Y/C Capture Mode .....	37
28	Utilization for Single Channel 8-Bit Y/C Capture Mode .....	38

29	Throughput for Single Channel 8-Bit Y/C Display Mode .....	39
30	Utilization for Single Channel 8-Bit Y/C Display Mode .....	40
31	Throughput for Single Channel 16-Bit Raw Video Display Mode .....	41
32	Utilization for Single Channel 16-Bit Raw Video Display Mode .....	42
33	3PSW Block Diagram.....	43
34	Transmit Buffer Descriptor .....	44
35	Receive Buffer Descriptor .....	45
36	Throughput for ESS With ICMP Protocol .....	48
37	Percentage of Utilization of ESS With ICMP .....	49
38	Throughput for ESS With TCP Protocol.....	50
39	Percentage of Utilization of ESS With TCP.....	51

### List of Tables

1	TMS320DM648/7 DMSoC Master Peripherals .....	4
2	TMS320DM648/7 DMSoC Slave Peripherals.....	5
3	System Connection Matrix .....	7
4	Default Burst Sizes .....	8
5	Memory Maximum Bandwidths.....	10
6	Default Master Priorities.....	10
7	Frequency and Bus Widths for Different Memory and Slave Endpoints .....	14
8	Factors Considered for Throughput .....	15
9	Read/Write Command Optimization Rules .....	19
10	EDMA3 Transfer Controller Configurations.....	20
11	EDMA Maximum Throughput for TC0 and TC2.....	24
12	EDMA Performance of EDMA for 8KB or 16KB Transfer.....	26
13	Factors Affecting McASP Throughput.....	27
14	Factors Considered for Throughput .....	31
15	Video Port Performances .....	31
16	Factors considered for throughput .....	46
17	ESS Performances.....	46

## 1 SoC Architectural Overview

The C64x+™ DSP megamodule, enhanced direct memory access (EDMA3) transfer controllers, and the system peripherals are interconnected through two switch fabrics. The switch fabrics allow for low-latency, concurrent data transfers between master peripherals and slave peripherals. Through a switch fabric, the central processing unit (CPU) can send data to the video ports without affecting a data transfer between the peripheral component interconnect (PCI) and the DDR2 memory controller. The switch fabrics also allow for seamless arbitration between the system masters when accessing system slaves. More information on SCR and bridges is provided in the following sections.

**Figure 1** shows the connection between slaves and masters through the data switched central resource (SCR). Masters are shown on the right and slaves on the left. The data SCR connects masters to slaves via 128-bit data buses running at a SYSCLK1 frequency. SYSCLK1 is supplied by the PLL1 controller and is fixed at a frequency equal to the CPU frequency divided by 3. Some peripherals, like PCI and the C64x+ megamodule, have both slave and master ports. Each EDMA3 transfer controller has an independent connection to the data SCR. Masters can access the configuration SCR through the data SCR.

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

Figure 2 shows the connection between the C64x+ megamodule and the configuration SCR, which is mainly used by the C64x+ megamodule to access peripheral registers. The data SCR also has a connection to the configuration SCR that allows masters to access most peripheral registers. The only registers not accessible by the data SCR through the configuration SCR are the device configuration registers and the PLL1 and PLL2 controller registers; these can be accessed only by the C64x+ Megamodule. The configuration SCR uses 32-bit configuration buses running at SYSCLK1 frequency.

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.
- The pattern of each arrow's line indicates the clock rate at which it is operating, at DSP/3, DSP/4 or DSP/6 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, PCI, Video and Imaging CoProcessor (VICP), Ethernet subsystem (ESS), VLYNQ™, and host port interface (HPI).
  - The peripheral/module has a master port as well as slave memories. Examples of these are the C64x+ megamodule and ESS.

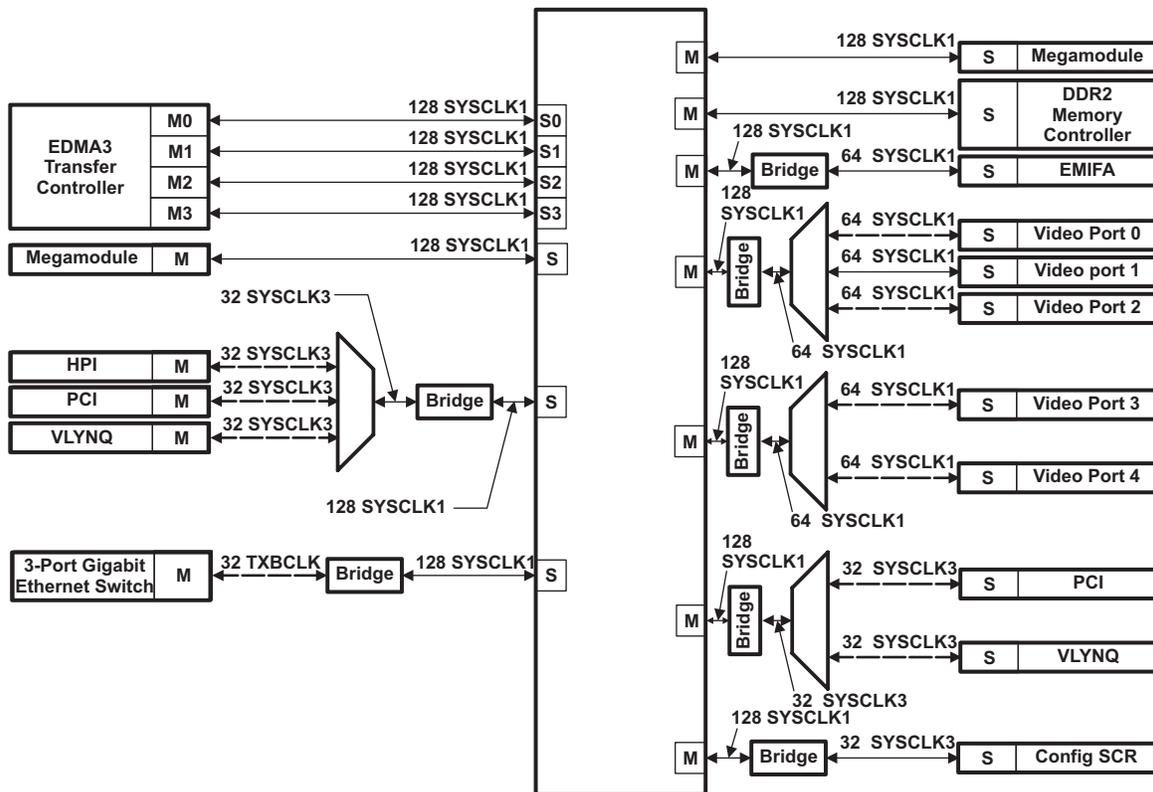


Figure 1. TMS320DM648/7 System Interconnect Block Diagram

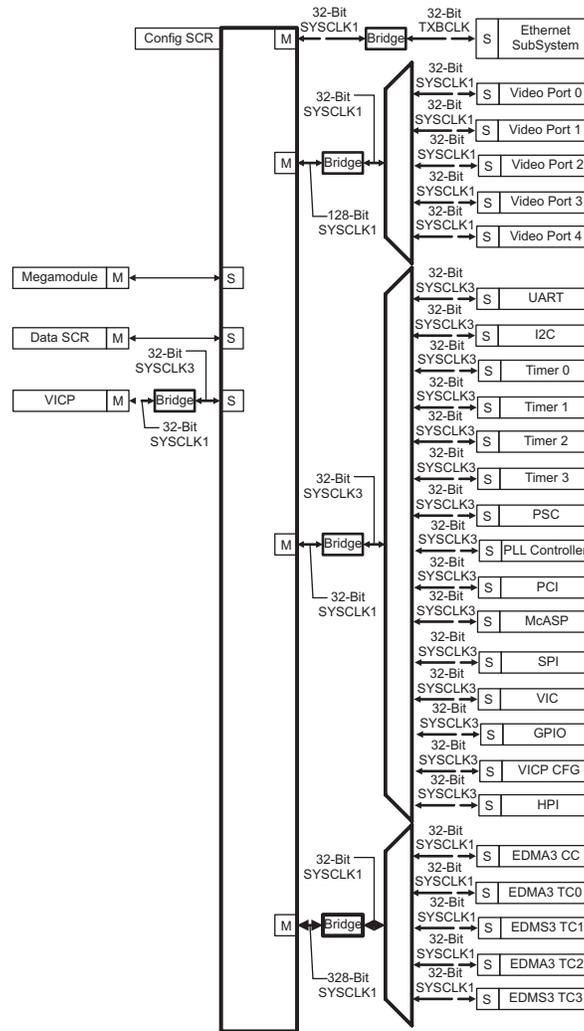


Figure 2. TMS320DM648/7 Peripheral Configuration Bus

### 1.1 Master Peripherals

The DM648/7 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 DM648/7 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 DM648/7 SoC master is also shown in Table 1.

Table 1. TMS320DM648/7 DMSoC Master Peripherals

Mstid	DM648/7 Master
0	EDMA TC0 Read Port
1	EDMA TC0 Write Port
2	EDMA TC1 Read Port
3	EDMA TC1 Write Port
4	EDMA TC2 Read Port
5	EDMA TC2 Write Port

**Table 1. TMS320DM648/7 DMSoC Master Peripherals (continued)**

Mstid	DM648/7 Master
6	EDMA TC3 Read Port
7	EDMA TC3 Write Port
8	ETHERNET SS
9	VLYNQ
10	HPI
11	PCI
13-Dec	Reserved
14	C64x+ CFG
15	C64x+ MDMA
16	VICP
17-63	Reserved

## 1.2 Slave Peripherals

Slave peripherals service the read/write transactions that are issued by master peripherals. All DM648/7 SoC slaves are listed in [Table 2](#). Note that memories are also classified as peripherals.

**Table 2. TMS320DM648/7 DMSoC Slave Peripherals**

DM648/7 Slaves
DDR2 Memory Controller
EMIFA
PCI Slave
C64x+ SDMA
VLYNQ Slave
VLYNQ Regs
EDMA3CC Regs
VideoPort 0
VideoPort 1
VideoPort 2
VideoPort 3
McASP

## 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 in the following sections. It is the decoding, routing, and arbitration logic that enables the connection between multiple masters and slaves that are connected to it. Multiple SCRs are used in the DM648/7 SoC, as shown in [Figure 1](#) and [Figure 2](#), to provide connections among different peripherals. Look at [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 DSP data (master) can send data to the DDR2 memory controller (slave) concurrently without affecting a data transfer between the PCI (master) and L2 memory (slave).

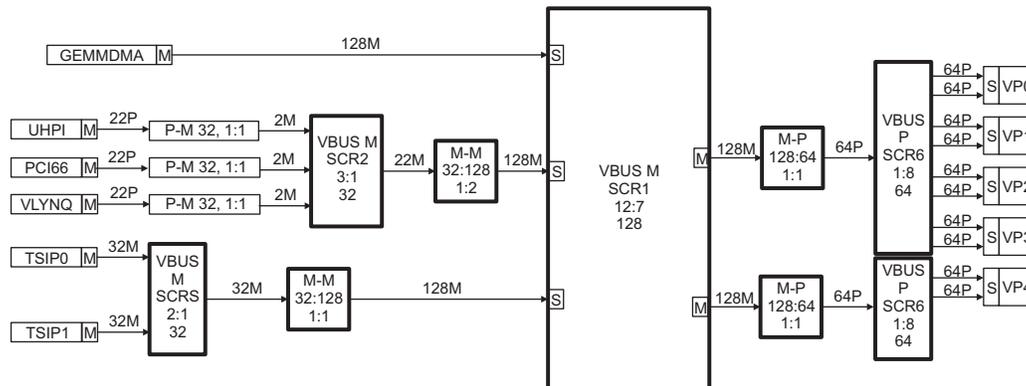
## 1.4 Bridge

Bridge In the DM648/7 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, there should be logic 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 3](#) shows the typical connection of a bridge.



**Figure 3. Bridge**

Multiple bridges are used in the DM648/7 SoC. For example, as shown in [Figure 4](#), 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 4. Bus-Width and Clock Rate Conversion**

### 1.4.1 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 5](#), 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.

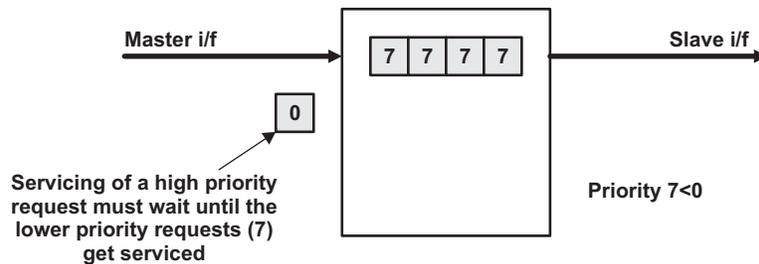


Figure 5. Bridge Head of Line Blocking

### 1.5 Master/Slave Connectivity

Not all masters on the device may connect to all slaves. Allowed connections are summarized in Table 3.

Table 3. System Connection Matrix

Masters	Slaves Figure 3										
	C64x+ SDMA	DDR2	EMIFB	VP0	VP1	VP2	VP3	VP4	VLYNQ	PCI	Config SCR
EDMA TC0 Read	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA TC0 Write	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA TC1 Read	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA TC1 Write	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA TC2 Read	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA TC2 Write	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA TC3 Read	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
EDMA TC3 Write	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
C64x+ MDMA	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
UHPI	Y	Y	Y	N	N	N	N	N	Y	Y	Y
PCI	Y	Y	Y	N	N	N	N	N	Y	Y	Y
VLYNQ	Y	Y	Y	N	N	N	N	N	N	N	N
Ethernet SS)	Y	Y	Y	N	N	N	N	N	N	N	N

### 1.6 Data Bus (Widths/Speeds)

There are two main types of busses on the DM648/7 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 DM648/7 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 receiver/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.

Table 4 shows default burst sizes of all the DM648/7 SoC masters.

**Table 4. Default Burst Sizes**

Master	Possible Burst Sizes (fixed or programmable)
C64x+ MDMA	
C64x+ CFG	
EDMA CC TR	
EDMA TC0 read/write	64 bytes (fixed)
EDMA TC1 read/write	
EDMA TC2 read/write	
EDMA TC3 read/write	
PCI	As slave: In slave mode, the burst size depends on the amount of data the external master requests and the available data bus band width at the time of transaction. As master: 64 bytes (fixed). Note that peripheral component interconnect (PCI) cannot burst more than 64 bytes in master mode.
HPI	Four 32-bit words (fixed)
VLYNQ	As Slave: N/A As Master: 64 bytes (fixed) (VLYNQ has no internal DMA, it relies on system EDMA for data transfer).
ESS	

## 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 PCI-to-DDR memory as shown in [Figure 6](#) (black line).

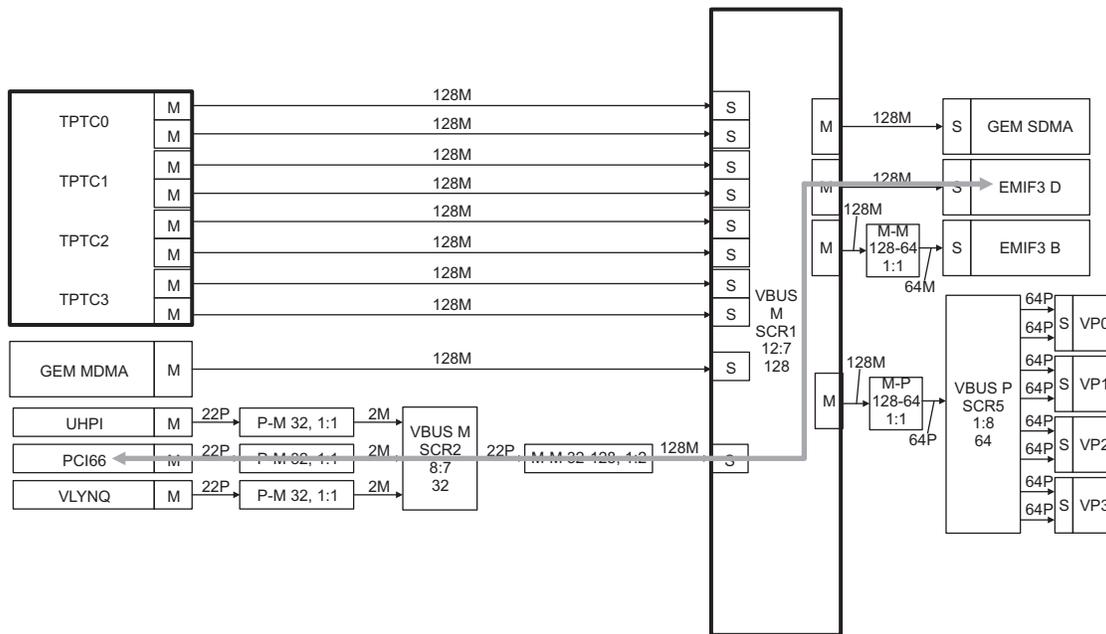


Figure 6. PCI-to-DDR Transfer

This transaction experiences latencies in the master PCI, SCR2, P-M Bridge, M-M Bridge, 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 SCR/bridges and experiences hardware latency. Polling on registers is more expensive.

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 hops from critical masters such as C64x+, and EDMA master to critical slaves such as C64x+ memory or DDR memory.

## 2.2 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, the master can issue a write transaction and go ahead with the next transaction without waiting for a response from the slave.

## 2.3 On-Chip Memory Vs Off-Chip Memory

On-chip memory access does not experience any hardware latency, whereas, off-chip memory access experiences hardware latency with SCR and bridges. 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 5](#) shows all memories of the DM648/7 SoC and their maximum bandwidths.

**Table 5. Memory Maximum Bandwidths**

Memory	Theoretical Maximum Bandwidth
C6x+ DSP L1P/L2 RAM	3840 MB/s (C64x+ SDMA frequency * C64x+ SDMA bus width = 240 MHz * 128-bit bus )
DDR2	2128 MB/s (DDR2 clock frequency * DDR2 bus width = 266 MHz * 64-bit bus)
EMIFB	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) 0.181818182

## 3 3 SoC Level Optimizations

This section describes system level optimization techniques.

### 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 *TMS320DM647/TMS320DM648 Digital Media Processors* ([SPRS372](#)). The default priority levels for the DM648/7 SoC bus masters are shown in [Table 6](#); lower values indicate higher priority.

**Table 6. Default Master Priorities**

Master	Default Priority
EDMA3TC0	0 <sup>(1)</sup>
EDMA3TC1	0 <sup>(1)</sup>
EDMA3TC2	0 <sup>(1)</sup>
EDMA3TC3	0 <sup>(1)</sup>
C64x+ (DMA)	7 <sup>(2)</sup>
C64x+ (CFG)	1
Ethernet SS	3
VLYNQ	4
UHPI	4
PCI	4
VICP	5

<sup>(1)</sup> Default value in EDMA3CC QUEPRI register

<sup>(2)</sup> Default value in C64x+ MDMAARBE.PRI field

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 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 *TMS320DM647/DM648 DSP DDR2 Memory Controller User's Guide* ([SPRUEK5](#)).

### 3.3 **C64x+ DSP Related Optimizations**

#### 3.3.1 **IDMA**

IDMA is optimized and best suited for L2 to L1D data transfers and vice versa. The intent of the IDMA is to offload CPU from on-chip memory (to/from L1D/L2) data movement tasks.

#### 3.3.2 **Choosing EDMA Vs CPU/IDMA**

The following are a few 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.

In summary, if concerned about L2 to L1 transfers, when geometry is fairly simple (i.e., 1-D transfer) 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 transfer), 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.

## 4 **IP Throughput**

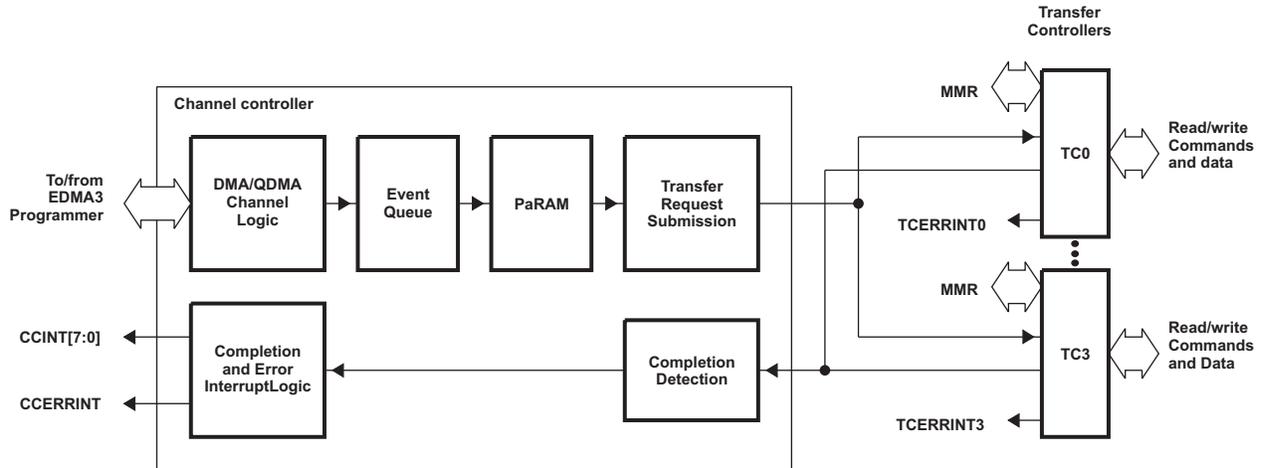
This section describes the maximum throughput performance of different peripherals of the DM648/7 SoC. It also provides the factors that affect peripheral throughput and recommendations for optimum peripheral performance.

### 4.1 **Enhanced Direct Memory Access (EDMA)**

This section provides a throughput analysis of the EDMA module integrated in the TMS320DM648/7 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 sub frame 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 7](#).

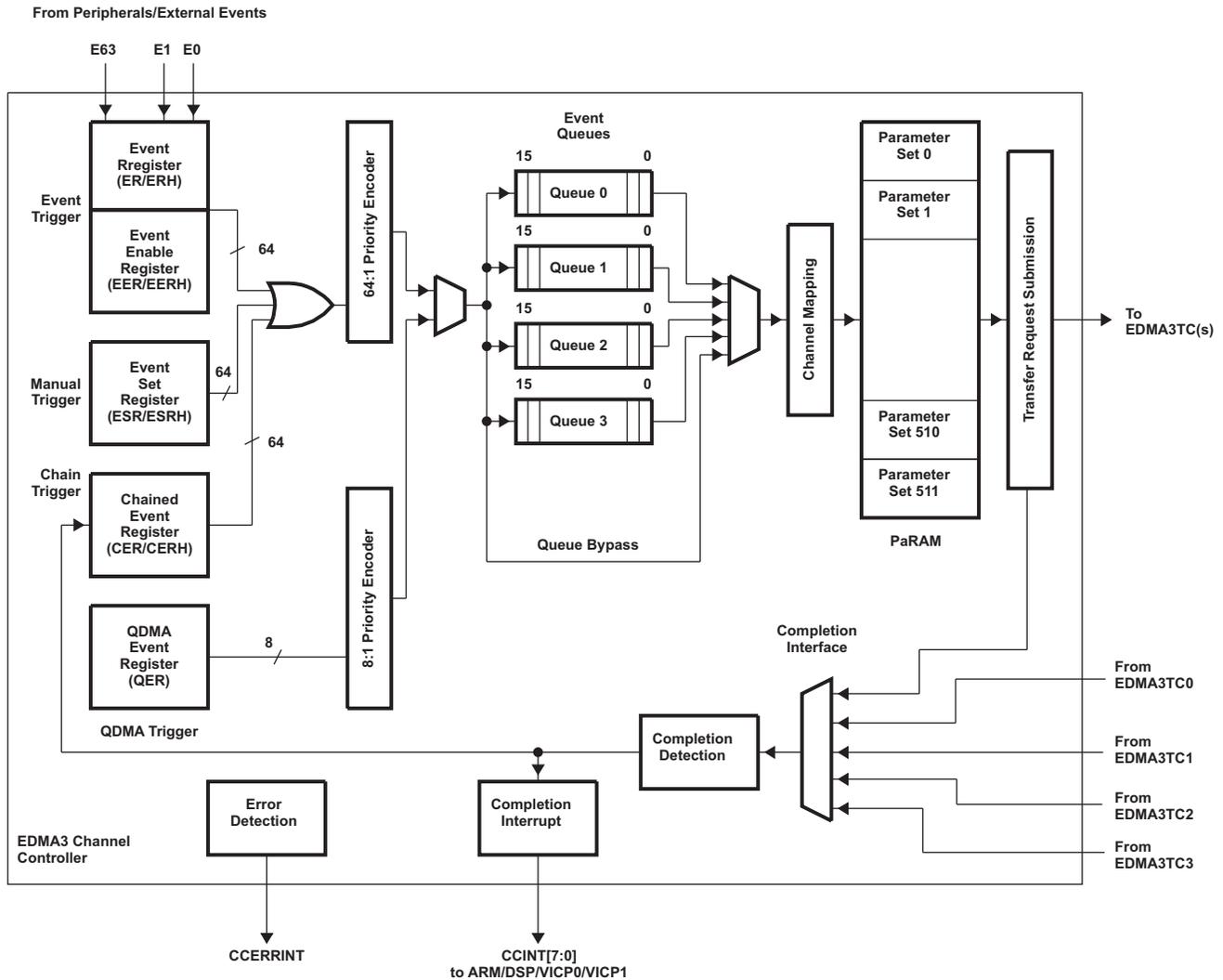


**Figure 7. EDMA3 Controller Block Diagram**

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

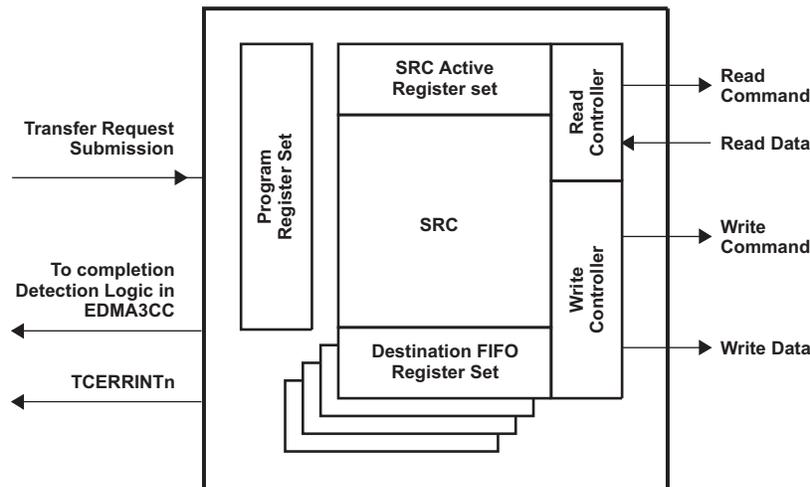
Once a trigger event is recognized, the event is queued in the programmed event queue. If two events are detected simultaneously, then the lowest-numbered channel has highest priority.

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 8** shows a block diagram of the channel controller.



**Figure 8. 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 9 shows a block diagram of the transfer controller.



**Figure 9. 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. The transfer never achieves the maximum theoretical bandwidth, due to the latency in the transmission. It is important to remember that latency will have a legacy impact for shorter transfers. Approximate latency for different memory accesses is equal to the time taken for a one byte transfer; this latency is not considered for throughput measurement in this document. Table 7 list the internal bus frequencies at which different memories and slave end point operates and their bus widths.

**Table 7. Frequency and Bus Widths for Different Memory and Slave Endpoints <sup>(1)</sup>**

Module Name	Freq (MHz)	Bus Width (bits)
DDR2	266	32
VICP		
GEM(L1P/L1D/L2) memories	450	
GEM Core	900	-

<sup>(1)</sup> The CC/TC of EDMA3 is operating at divide-by 3 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: 900 MHz
- DDR clock: 266 MHz
- Used DSP's internal timer operating at 900 MHz
- Throughput data collected is standalone. No other ongoing 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 8](#).

**Table 8. Factors Considered for Throughput**

<b>Factors</b>	<b>Impact</b>	<b>Recommendations</b>
Source/Destination Memory	The transfer speed depends on SRC/DST memory bandwidth	Use L1, L2, or DDR for better results. Avoid AEMIF.
Transmit Size	Throughput depends on 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	Configure BIDX equal to ACNT value
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 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	Performance degrades if there is a mismatch in alignment	Set source destination alignment value to zero for better performance
CPU and DDR Frequency	The utilization is above 90a5 for both CPU and DDR frequency variation	

### 4.1.4 Transfer Size

Throughput is low for smaller transfer size (less than 1Kbytes) due to transfer overhead. Large transfer sizes give higher throughput. Figure 10 describes the percentage of utilization for transfers between L2 and DDR with different transfer sizes. Figure 11 describes throughput for transfers between L2 and DDR with different transfer sizes.

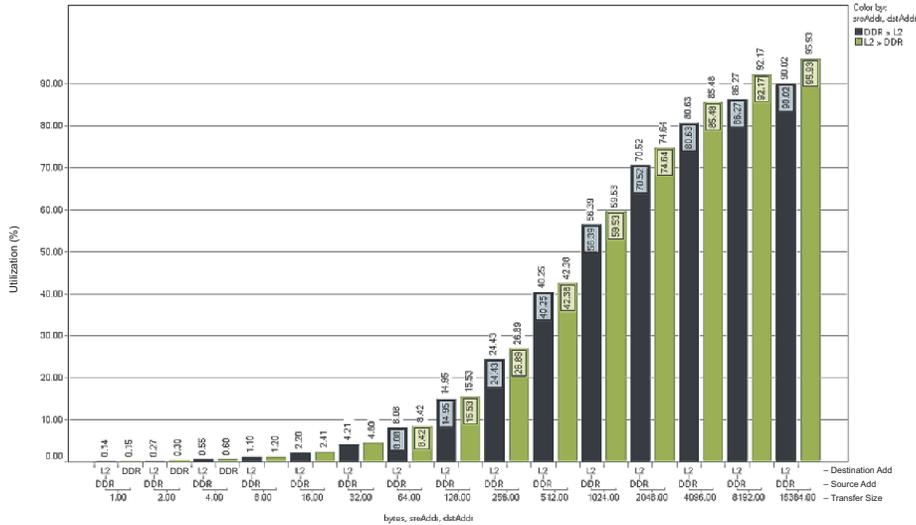


Figure 10. Utilization of EDMA for L2, DDR Access

As transfer size increases, the impact of the transfer overhead/latency reduces, therefore, the percentage of utilization also increases. 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.

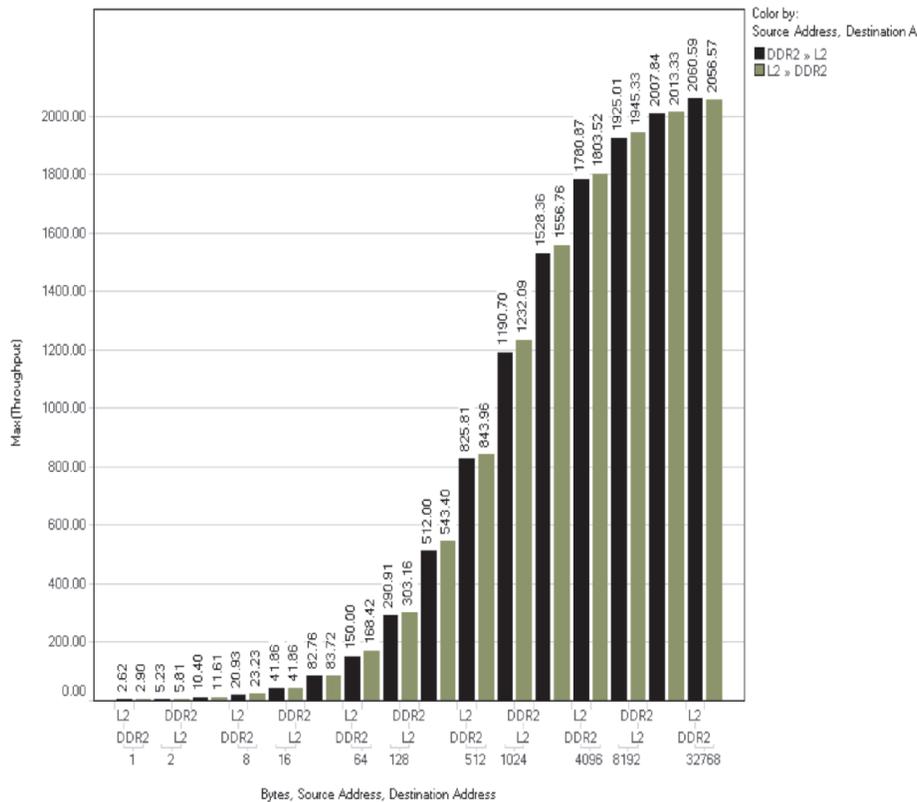


Figure 11. Throughput of EDMA for L2, DDR Access

As transfer size increases, the impact of the transfer overhead/latency reduces, therefore, the throughput also increases. 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 throughput is above 1100 Mbytes per sec for a transfer with a data size greater than 1kB.

#### 4.1.5 A-Sync/AB-Sync

An A-sync transfer is configured as follows:

- The number of TRs 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 that reduces transfer overhead.

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

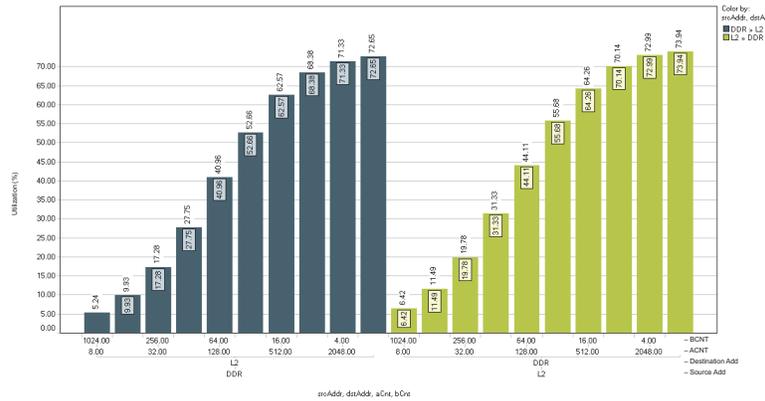


Figure 12. Utilization for Different Element Size (ACNT)

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.

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

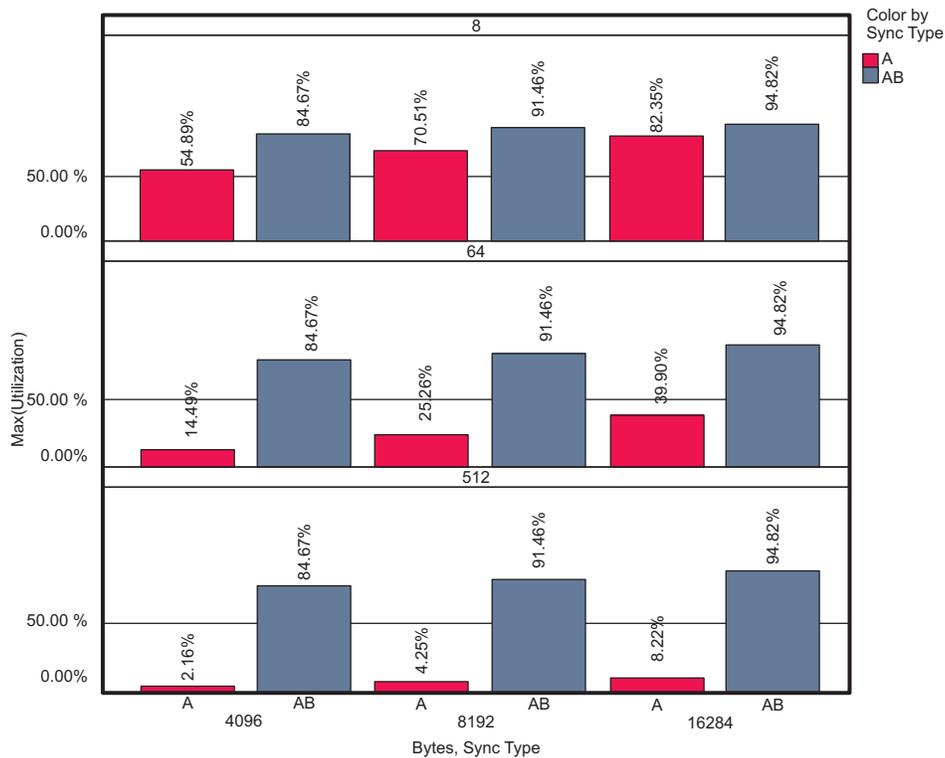


Figure 13. Effect of A-Sync and AB-Sync

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.

#### 4.1.6 TC Optimization Rules

If  $ACNT \leq DBS$  (default burst size) and ACNT is power of 2 and  $SRCDIDX/DSTBIDX = ACNT$  and  $BCNT \leq 1023$  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 9](#).

**Table 9. Read/Write Command Optimization Rules**

ACNT $\leq$ DBS	ACNT is Power of 2	BIDX = ACNT	BCNT $\leq$ 1023	SAM/DAM Increment	Description
Yes	Yes	Yes	Yes	Yes	Optimized
No	X	X	X	X	Not Optimized
X	No	X	X	X	Not Optimized
X	X	No	X	X	Not Optimized
X	X	X	No	X	Not Optimized

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.

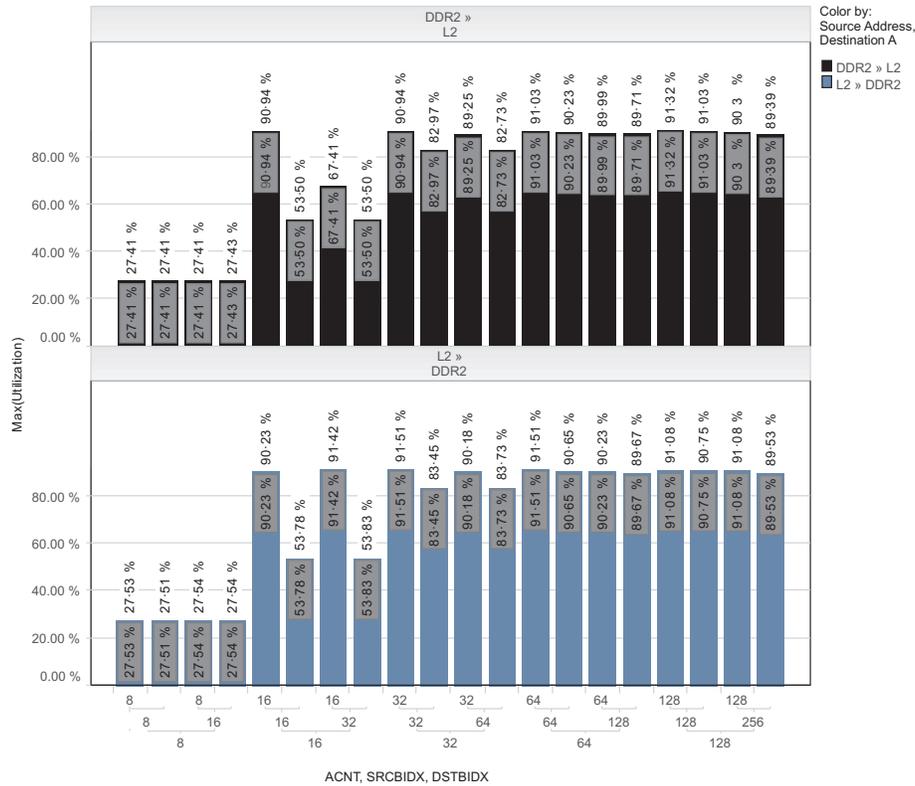


Figure 14. Utilization for Different Destination Index Value

In Figure 14, Y-axis represents the percentage of utilization 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. 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.

#### 4.1.7 Queue TC Usage

On DM647/8, there are four transfer controllers to move data between slave end points. The default configuration for the transfer controllers is shown in Table 10.

Table 10. EDMA3 Transfer Controller Configurations

Name	TC0	TC1	TC2	TC3
FIFOSIZE	128 Bytes	128 Bytes	512 Bytes	512 Bytes
BUSWIDTH	16 Bytes	16 Bytes	16 Bytes	16 Bytes
DSTREGDEPTH	2 entries	4 entries	4 entries	4 entries
Default DBS	64 Bytes	64 Bytes	64 Bytes	64 Bytes

The individual TC performance for paging/memory to memory transfers is essentially dictated by the TC configuration. In most scenarios, the FIFOSIZE configurations 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 DM647/8 device, TC0, TC1 transfer controllers yield identical performance for all transfer scenarios and TC2, TC3 transfer controllers yield identical performance for all transfer scenarios [Figure 15](#) and [Figure 16](#) shows the throughput of TC0, TC1 and TC2, TC3 respectively.

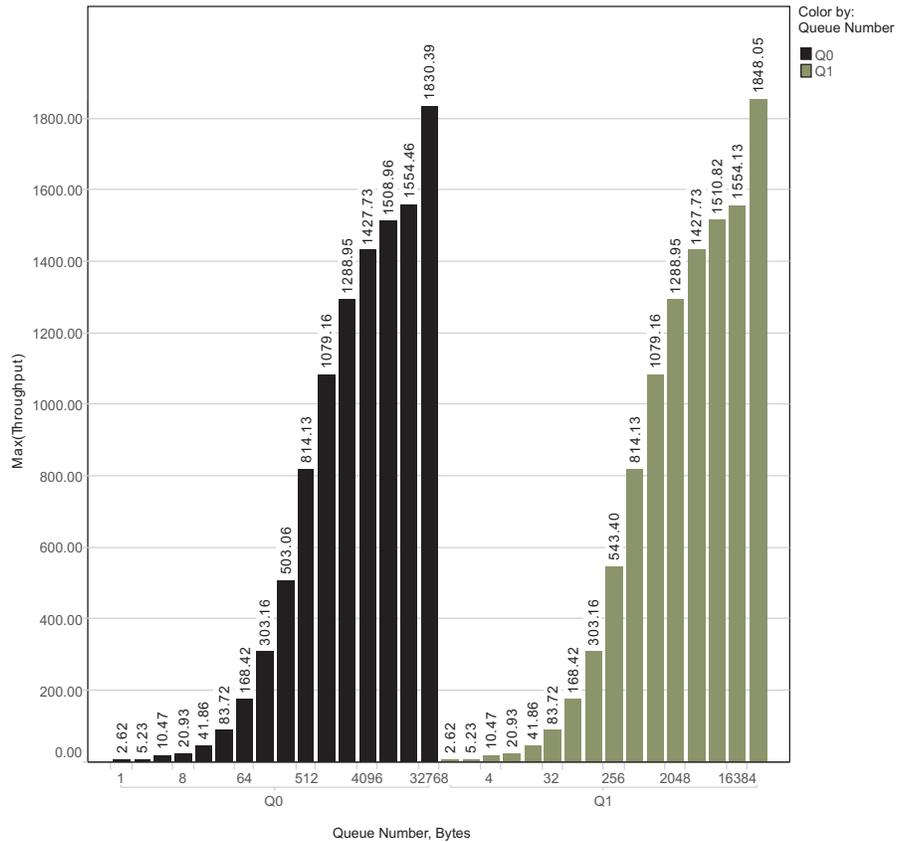
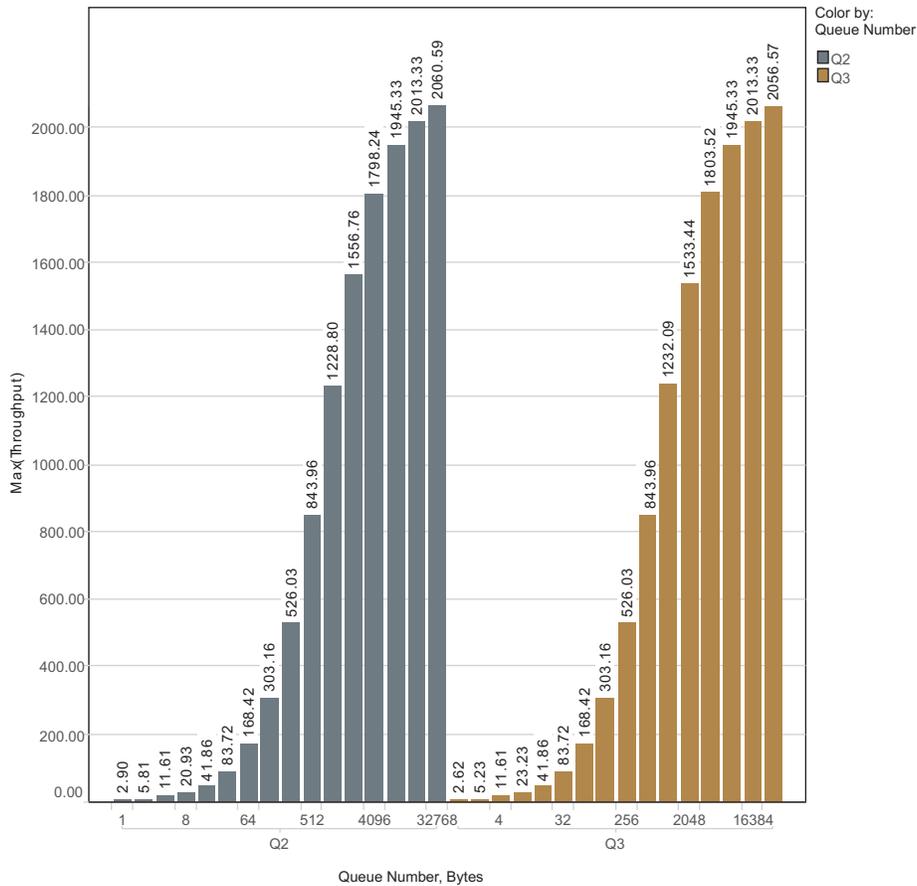


Figure 15. Performance of TC0 and TC1



**Figure 16. Performance of TC0 and TC1**

The Y-axis represents the throughput in MBps and the X-axis represents the different transfer size for TC0, TC1, TC2 and TC3. On the DM647/8 device, TC0, TC1 transfer controllers yield identical performance for all transfer scenarios and TC2, TC3 transfer controllers yield identical performance for all transfer scenarios.

**4.1.8 Burst Size**

The DBS on DM647/8 is fixed to 64 bytes; programmability of DBS on DM647/8 is not supported.

### 4.1.8.1 CPU and DDR Frequency Variation

Figure 17 and Figure 18 show the performance of EDMA for transfer between L2 and DDR for 16KB and 32KB transfer size.

The maximum throughput in MBps and Utilization in % for TC0 and TC2 are shown in Table 11.

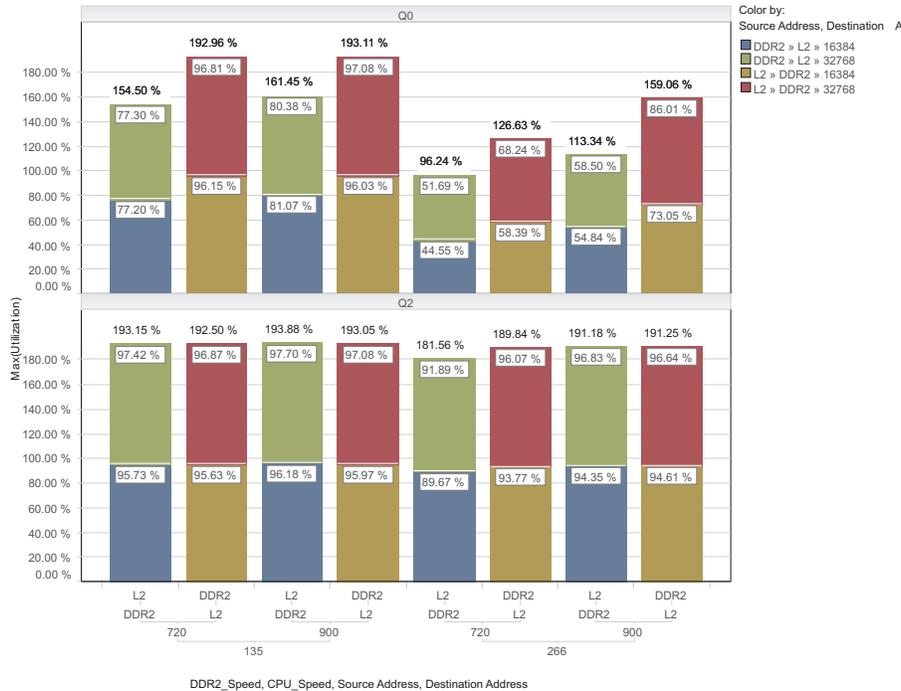


Figure 17. Utilization for TC0 and TC2 for Different CPU and DDR Frequency

The Y-axis represents the percentage of utilization for transfer between DDR2 to L2 and L2 to DDR2 and the X-axis represents different CPU frequency configuration for given DDR frequency.

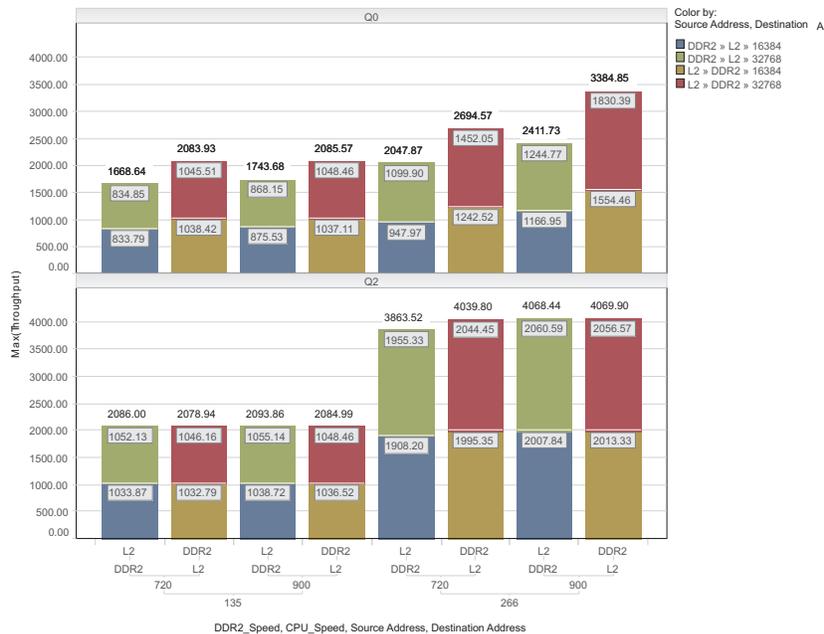


Figure 18. Max throughput for TC0 and TC2 for Different CPU and DDR Frequency

The Y-axis represents the maximum throughput for transfer between DDR2 to L2 and L2 to DDR2 and the X-axis represents different CPU frequency configuration for given DDR frequency.

**Table 11. EDMA Maximum Throughput for TC0 and TC2**

EDMA TC	DDR2 Frequency (in MHz)	CPU Frequency (in MHz)	Transfer Size (in KB)	Source Address	Destination Address	Utilization (in %)	Throughput (in MBps)
TC0	135	720	16	DDR2	L2	77.2	833.79
				L2	DDR2	96.15	1038.42
		900	DDR2	L2	81.07	875.53	
			L2	DDR2	96.03	1037.11	
	266	720		DDR2	L2	44.55	947.97
				L2	DDR2	58.39	1242.52
		900	DDR2	L2	54.84	1166.95	
			L2	DDR2	73.05	1554.46	
	135	720	32	DDR2	L2	77.3	834.85
				L2	DDR2	96.81	1045.51
		900	DDR2	L2	80.38	868.15	
			L2	DDR2	97.08	1048.46	
		266	720	DDR2	L2	51.69	1099
				L2	DDR2	68.24	1452.05
TC2	135	720	16	DDR2	L2	58.5	1244.77
				L2	DDR2	86.01	1830.39
		900	DDR2	L2	95.73	1033.87	
			L2	DDR2	95.63	1032.79	
	266	720		DDR2	L2	96.18	1038.72
				L2	DDR2	95.97	1036.52
		900	DDR2	L2	89.67	1908.2	
			L2	DDR2	93.77	1995.35	
	135	720	32	DDR2	L2	94.35	2007.84
				L2	DDR2	94.61	2013.33
		900	DDR2	L2	97.42	1052.13	
			L2	DDR2	96.87	1046.16	
		266	720	DDR2	L2	97.7	1055.14
				L2	DDR2	97.08	1048.46
	900	720		DDR2	L2	91.89	1955.33
				L2	DDR2	96.07	2044.45
		900	DDR2	L2	96.83	2060.59	
			L2	DDR2	96.64	2056.57	

#### 4.1.8.2 Performance of EDMA

Figure 19, Figure 20 and Table 12 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.

Figure 20 shows throughput on Y-axis and various source and destination memory combinations on X-axis, color coded for extra clarity. Table 12 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 16kB, BCNT and CCNT equal to 1, A-Sync transfers with increment addressing mode and CPU/DDR/memory setup as specified in 1.1.2.

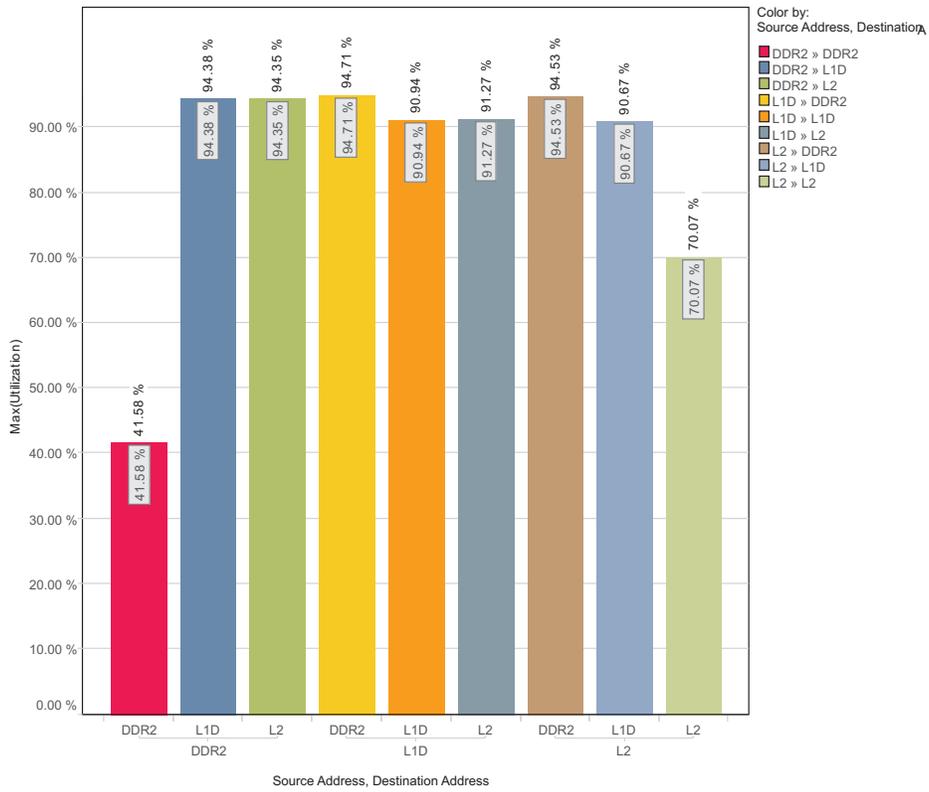


Figure 19. EDMA Performance

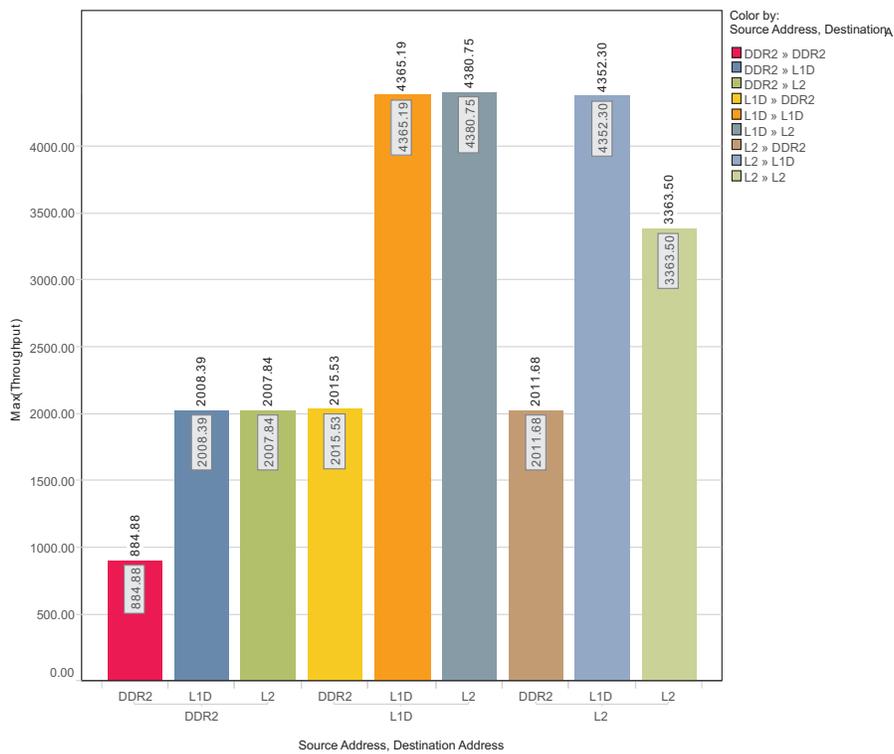


Figure 20. EDMA Performance

**Table 12. EDMA Performance of EDMA for 8KB or 16KB Transfer**

Source Mem	Destination Mem	Actual Throughput (MBytes/sec)	Theoretical Maximum Throughput (MBytes/sec)	Utilization (%)	Xfer Size (in KB)
DDR2	DDR2	884.88	2128	41.48	16
	L1D	2008.39	2128	94.38	16
	L2	2007.84	2128	94.35	16
L1D	DDR2	2015.53	2128	94.71	16
	L1D	4365.19	4800	90.94	16
	L2	4380.75	4800	91.27	16
L2	DDR2	2011.68	2128	94.53	16
	L1D	4352.3	4800	90.67	16
	L2	3363	4800	70.07	16

## 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 inter component 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.

The McASP module includes up to ten serializers that can be individually enabled to either transmit or receive in all different modes.

### 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 21](#), 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 PLL1 clock divided by 6. The McASP serial clock (clock at the bit rate) can be sourced from:

- Internally: passing through two clock dividers off the AUX\_CLKIN (SYSCLK3) clock
- Externally: directly from the ACLKR/ACLKX pin
- Mixed: an external clock is input to the McASP on either the AHCLKX or HCLKR 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 *TMS320DM647/DM648 DSP Multichannel Audio Serial Port (McASP) User's Guide (SPRUEL1)*.

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 25 MHz, obtained by setting the appropriate two clock dividers value for a given CPU frequency. Therefore, for each serializer, the theoretical maximum throughput is 25 Mbps, regardless of receiving or transmitting (when the clock is 25 MHz). When all the serializers are activated (McASP has ten serializers), the theoretical maximum throughput is 25 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: 900 MHz
- DDR clock rate: 266 MHz
- AEMIF configuration
  - Read time cycle (setup/strobe/hold): 22 (4/16/2)
  - Write time cycle (setup/strobe/hold): 22 (4/16/2)
  - Data bus width: 16 bits
- McASP serial clock mode: sourced internally
- McASP master clock (AHCLKX/AHCLKR) rate: 25 MHz (set HCLKRDIV/HCLKXDIV to 5)
- Only two serializers are active during the analysis: one for transmitting and one for receiving
- This is a standalone McASP throughput analysis; the numbers might vary when additional peripherals are competing for system resources.
- McASP is configured to transfer 8 bits, 16 bits and 32 bits and EDMA is configured for 32 bits transfer per McASP event irrespective of McASP element transfer (8/16/32).

#### 4.2.5 Factors Affecting McASP Throughput

[Table 13](#) lists the factors that might affect McASP throughput.

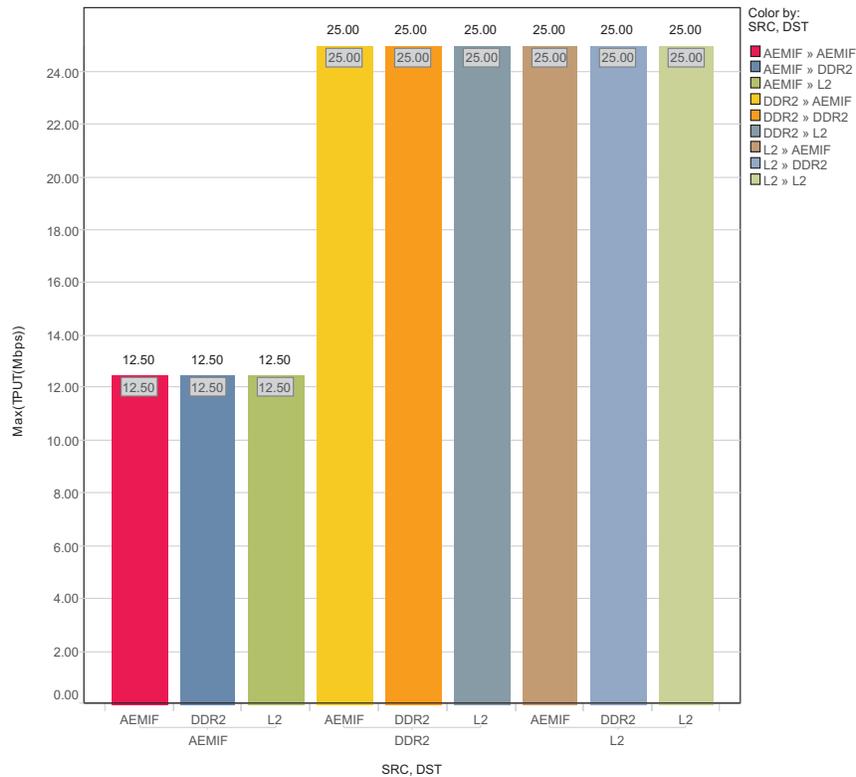
**Table 13. Factors Affecting McASP Throughput**

<b>Factor</b>	<b>Impact</b>	<b>General Recommendation</b>
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 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.

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-bit, and 8-bit element scenarios.

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

Figure 21 shows the throughput analysis for the 16- and 32-bit element.



**Figure 21. 16- and 32-Bit Element Throughput Analysis**

#### 4.2.6 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 25 Mbps throughput. When the SRC buffer is in AEMIF memory, McASP can only maintain accurate transfer at 12.5 MHz bit clock regardless of the DST buffer location.

#### 4.2.7 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 25 MHz, which in turn produces a throughput equal to the theoretical number.

### 4.3 VIDEO PORT

This section provides the throughput analysis of the video port module integrated in the TMS320DM647/648 SoC.

### 4.3.1 Overview

Video port is capable of sending and receiving digital video data. The video ports are also capable of capturing/displaying RAW data. The video port peripherals follow video standards such as BT.656 and SMPTE296. The video port peripheral can operate as a video capture port, video display port, or transport channel interface (TCI) capture port.

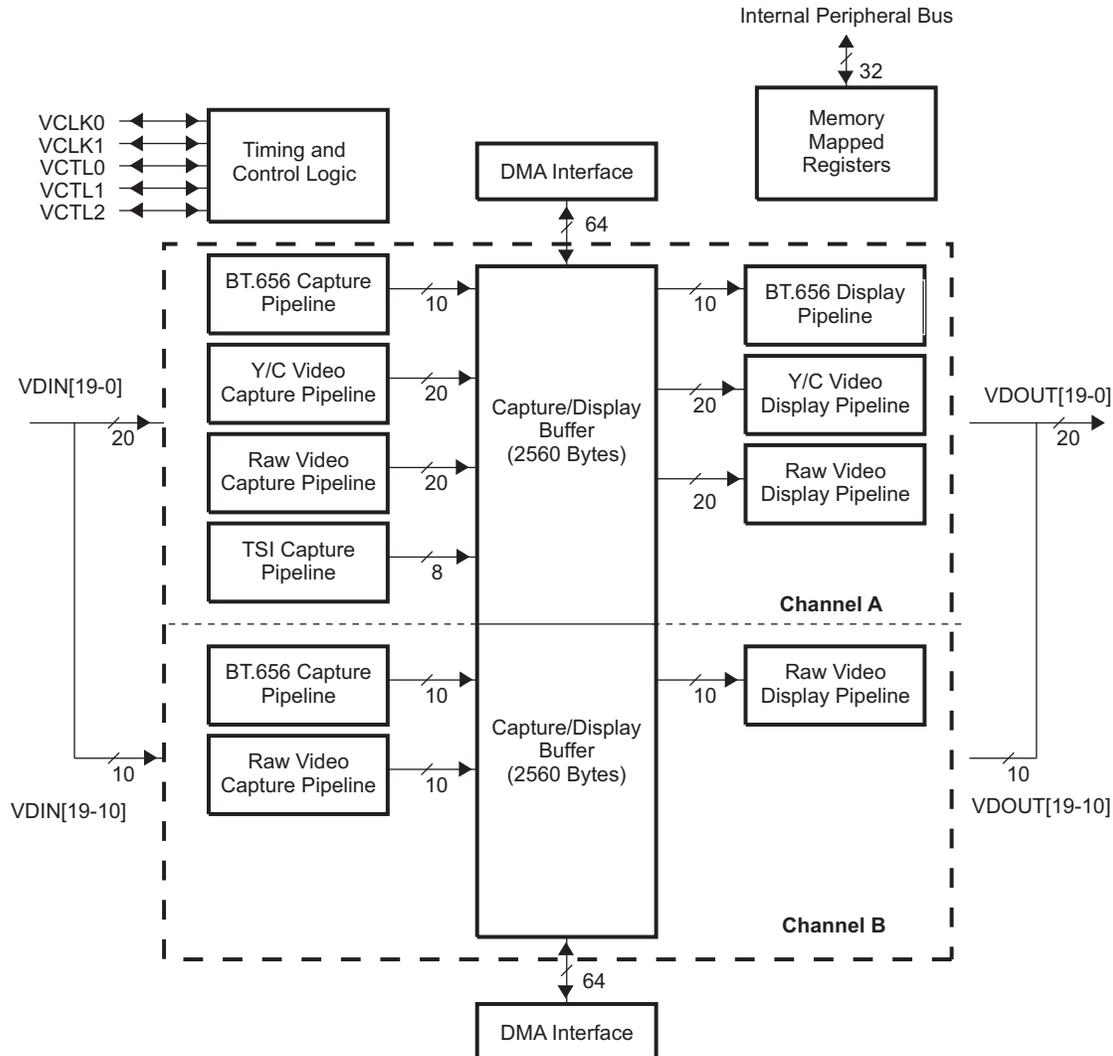


Figure 22. Video Port Functional Block Diagram

The port consists of two channels, A and B. 5120-byte capture/display buffer is splittable between the two channels. The entire port (both the channels) is always configured for either video capture or display only. Separate data pipelines control the parsing and formatting of video capture or display data for each of the BT.656, Y/C, raw video, and TCI modes. Channel B is not used during single channel operation.

For video capture operation, the video port may operate as:

- Two 8-bit channels of BT.656
- Two 8-bit channels of raw video
- Single 8-bit channel of BT.656
- Single 8-bit channel of raw video
- Single 8-bit channel Y/C video
- Single 16-bit channel of raw video
- Single 8-bit channel of TCI

For video display operation, the video port may operate as:

- Single 8-bit channel of BT.656
- Single 8-bit channel of raw video
- Single 8-bit channel of Y/C video
- Single 16-bit channel of raw video

TCI capture mode from front-end device (such as demodulator) or a forward error correction device in 8-bit parallel format at up to 30 Mbytes/sec.

#### 4.3.2 Video Port FIFO

The video port includes a FIFO to store data coming into or out from the video port. The video port operates in conjunction with EDMA transfers to move data between the video port FIFO and external or on-chip memory. EDMA events are generated when the video port FIFO reaches certain fullness (for capture) or goes below certain fullness (for display).

#### 4.3.3 EDMA Operation

The video port uses up to three EDMA events per channel for a total of six possible events. Each EDMA event uses a dedicated event output.

##### 4.3.3.1 Capture EDMA Event Generation

If no EDMA event is currently pending and the FIFO crosses the value specified by threshold, an EDMA event is generated. Once an event has been requested, another EDMA event may not be generated until the servicing of the outstanding event has begun. For BT.656 and Y/C modes, there are three FIFOs, one for each of the Y, Cb and Cr color components. Each FIFO generates its own EDMA event, therefore, the EDMA event state and FIFO thresholds for each FIFO are tracked independently. The Cb and Cr FIFOs use threshold value of the threshold value of Y.

##### 4.3.3.2 Display EDMA Event Generation

Display EDMA events are generated based on the amount of room available in the FIFO. The threshold value indicates the level at which the FIFO has room to receive another EDMA. If the FIFO has at least threshold level location available, an EDMA event is generated. Once an EDMA event has been requested, another EDMA event may not be generated until the servicing of the first EDMA event has begun. For BT.656 and Y/C modes, there are three FIFOs, one for each of the Y, Cb and Cr color components. Each FIFO generates its own EDMA event, therefore, the EDMA event state and FIFO thresholds for each FIFO are tracked independently. The Cb and Cr FIFOs use threshold value of threshold value of Y.

#### 4.3.4 Video Capture Port

In video capture mode, capture rate is up to 80 MHz. Video capture works by sampling video data on the input pins and saving it to the video port FIFO. When the amount of captured data reaches a programmed threshold level, an EDMA is performed to move data from the FIFO into the DSP memory. In some cases, color separation is performed on the incoming video data requiring multiple FIFOs and EDMAs to be used. The video port supports capture of both interlaced and progressive scan data. Interlaced capture can be performed on either a field-by-field or a frame-by-frame basis. A capture window specifies the data to be captured within each field. Frame and field synchronization can be performed using embedded sync codes or configurable control inputs allowing glueless interface to various encoders and ADCs.

#### 4.3.5 Video Display Port

In video display mode, display rate is up to 110 MHz. Video display works by moving data from video port FIFO to the output pins. When there is at least the threshold number of double word free in the FIFO, EDMA is performed to move data from DSP memory to video port FIFO. Video port supports display of both interlaced and progressive scan data.

The formulas used for throughput calculations are given below:

- Ideal Throughput = Frequency of pixel clock \* data bus width in bytes.
- Actual Throughput = (FPS \* data bus width in bytes \* width in pixels \* height in pixels).
- Bandwidth Utilization = (Actual Throughput / Ideal Throughput) \* 100

#### 4.3.6 Factors Affecting Videoport Throughput Value

To gain high throughput and the best performance, videoport has to be configured properly.

The different factors considered for throughput calculation with its impact is given in [Table 14](#)

**Table 14. Factors Considered for Throughput**

Factors	Impact	Recommendation
EDMA Count	Performance depends on the number of EDMA events generated. More events means more overhead.	EDMA count should be equal to one line in the image so that EDMA event is generated for every line of the image.
FIFO SIZE	If the threshold value of FIFO is not configured properly, it leads to underflow/ overrun.	Always threshold value of the FIFO as to be configured equal to the EDMA count.
Data width	Only in raw video mode (both capture and display mode). Throughput for 8-bit is half the throughput value when it is compared to 16-bit.	
Pixel clock	Only in raw video mode (both in capture and display). Throughput value varies, when pixel clock is varied with fixed FPS and image size.	

#### 4.3.7 Performance of Video Port

[Table 15](#) captures the throughput and percentage utilization for different video standards.

**Table 15. Video Port Performances**

Display/ Capture	Standard	Field Mode ( <sup>1</sup> )	VPORT Mode	(MHz) Pixel Clock	FPS	Width in Pixels	Height in Pixels	Actual Tput MB/sec	TC Used	VPORT	Ideal Tpu MB/sect	Utilization
Display	NTSC	I	8-bit BT.656	27	30	720	480	19.78	3	VP1	27	73.26%
Display	PAL	I	8-bit BT.656	27	25	720	576	19.78	3	VP1	27	73.26%
Capture	NTSC	I	8-bit BT656	27	30	720	480	158.24	2	VP0,2,3,4	216	73.26%
Capture	PAL	I	8-bit BT656	27	25	720	576	158.24	2	VP0,2,3,4	216	73.26%
Display	HDTV 720p 60Hz	P	8-bit Y/C	74.25	60	1,280	720	105.47	3	VP1	148.5	71.02%
Display	HDTV 1080i 60Hz	I	8-bit Y/C	74.25	30	1,920	1,080	118.65	3	VP1	148.5	79.90%
Display	HDTV 1080i 50Hz	I	8-bit Y/C	74.25	25	1,920	1,080	98.88	3	VP1	148.5	66.59%
Capture	HDTV 720p 60Hz	P	8-bit Y/C	74.25	60	1,280	720	105.47	2	VP0	148.5	71.02%
Capture	HDTV 1080i 60Hz	I	8-bit Y/C	74.25	30	1,920	1,080	118.65	2	VP0	148.5	79.90%
Capture	HDTV 1080i 50Hz	I	8-bit Y/C	74.25	25	1,920	1,080	98.88	2	VP0	148.5	66.59%
Display	VGA 60Hz	P	16-bit RAW	25.18	60	640	480	35.16	3	VP1	50.36	69.82%
Display	VGA 72Hz	P	16-bit RAW	31.5	72	640	480	42.19	3	VP1	63	66.97%

(<sup>1</sup>) Where I = Interlaced, P = Progressive

**Table 15. Video Port Performances (continued)**

Display/ Capture	Standard	Field Mode (1)	VPORT Mode	(MHz) Pixel Clock	FPS	Width in Pixels	Height in Pixels	Actual Tput MB/sec	TC Used	VPORT	Ideal Tpu MB/sect	Utilization
Display	VGA 75Hz	P	16-bit RAW	31.5	75	640	480	43.95	3	VP1	63	69.76%
Display	VGA 85Hz	P	16-bit RAW	36	85	640	480	49.8	3	VP1	72	69.17%
Display	SVGA 60Hz	P	16-bit RAW	40	60	800	600	54.93	3	VP1	80	68.66%
Display	SVGA 72Hz	P	16-bit RAW	50	72	800	600	65.92	3	VP1	100	65.92%
Display	SVGA 75Hz	P	16-bit RAW	49.5	75	800	600	68.66	3	VP1	99	69.35%
Display	SVGA 85Hz	P	16-bit RAW	56.25	85	800	600	77.82	3	VP1	112.5	69.17%
Display	XGA 60Hz	P	16-bit RAW	65	60	1,024	768	90	3	VP1	130	69.23%
Display	XGA 70Hz	P	16-bit RAW	75	70	1,024	768	105	3	VP1	150	70.00%
Display	SXGA 60Hz	P	16-bit RAW	108	60	1,280	1,024	150	3	VP1	216	69.44%

### 4.3.8 Video Port Configured as BT.656 Capture Mode

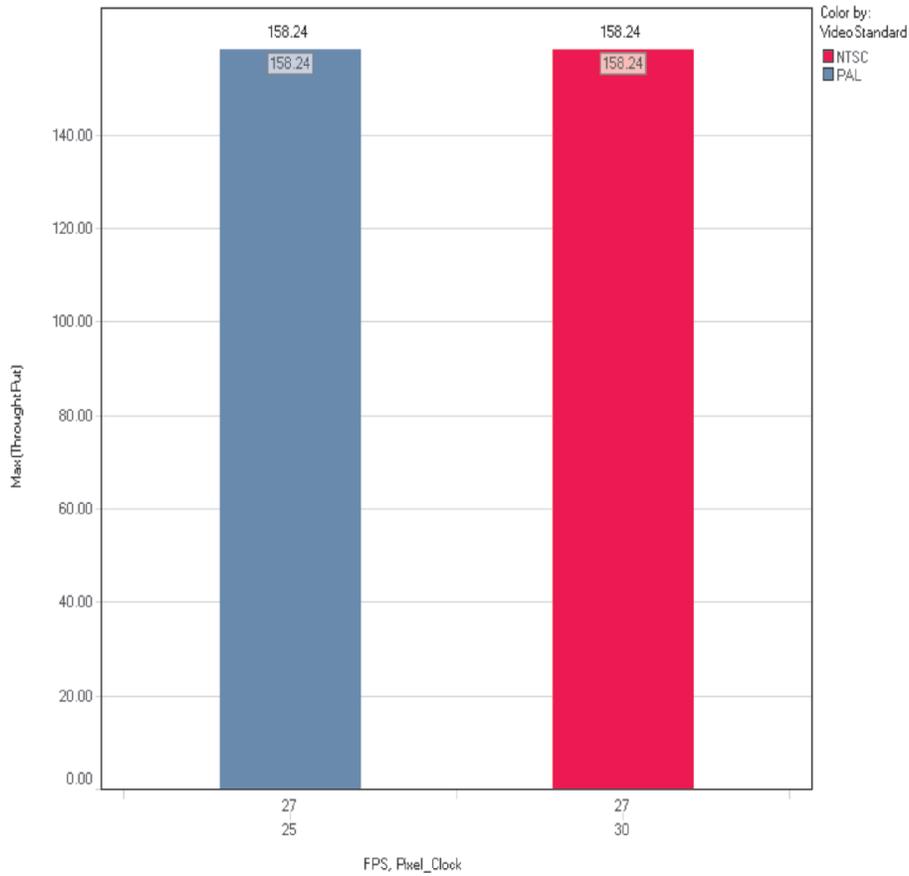
#### 4.3.8.1 Test Environment

System setup for 8 captures (8-bit BT.656 on channel A and B for each videoport 0, 2, 3, and 4) throughput measurement is given below:

- CPU clock: 900 MHz.
- DDR2 clock: 266 MHz.
- Video Port: Port 0, 2, 3 and 4
- EDMA Transfer Controller: TC2
- Video port mode: 8 captures (8-bit BT.656 on channel A and B for each video port 0, 2, 3, and 4)
- Throughput data collected is standalone. No other ongoing traffic.
- Throughput data collected using NDK drivers.

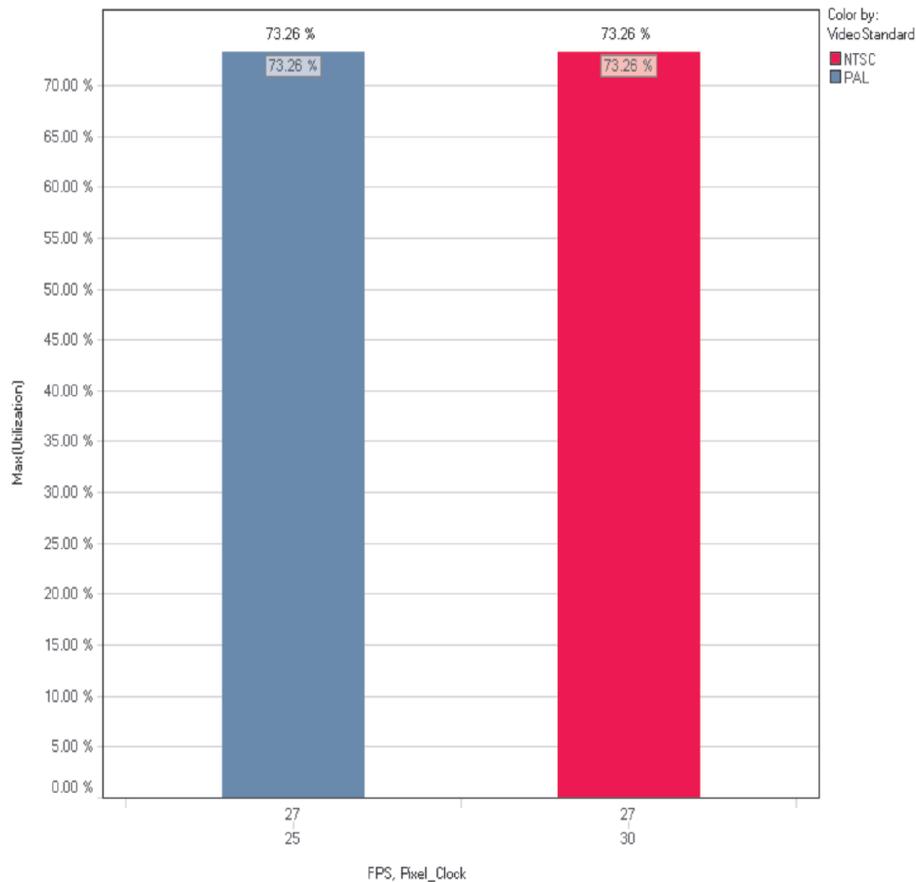
**4.3.8.2 Performance of Video Port**

Figure 23 shows the throughput for 8 captures in 8-bit BT.656 capture mode on channel A and B. The Y-axis represents the throughput in MB/sec and X-axis represents the different FPS for fixed 27 MHz pixel clock.



**Figure 23. Throughput for 8 Captures in 8-bit BT.656 Capture Mode**

Figure 24 shows the percentage utilization for 8 captures in 8-bit BT.656 capture mode on channel A and B. The Y-axis represents percentage utilization and X-axis represents the different FPS for fixed 27 MHz Pixel clock.



**Figure 24. Utilization for 8 Captures in 8-bit BT.656 Capture Mode**

### 4.3.9 Video Port Configured as BT.656 Display Mode

#### 4.3.9.1 Test Environment

System setup for single channel 8-bit BT.656 display mode throughput measurement is given below:

- CPU clock: 900 MHz.
- DDR2 clock: 266 MHz.
- Video Port: Port 1
- EDMA Transfer Controller: TC3
- Video port mode: Single channel 8-bit BT.656 video display
- Throughput data collected is standalone. No other ongoing traffic.
- Throughput data collected using NDK drivers.

### 4.3.9.2 Performance of Video Port

Figure 25 shows the throughput for single channel 8-bit BT.656 display mode. The Y-axis represents the throughput in MB/sec and X-axis represents the different FPS for fixed 27 MHz pixel clock.

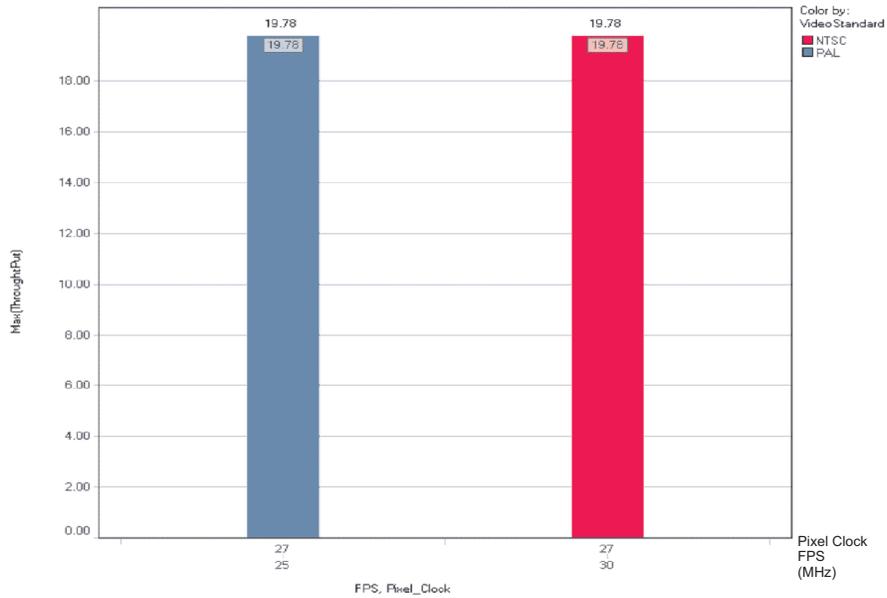
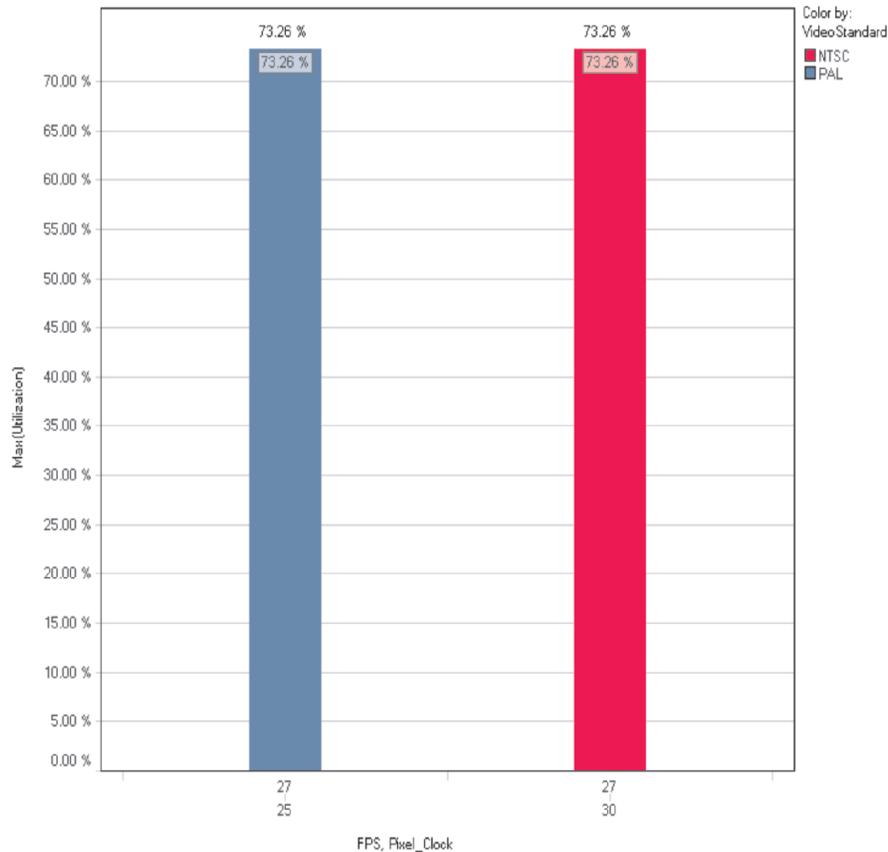


Figure 25. Throughput for Single Channel 8-Bit BT.656 Display Mode

Figure 26 shows the percentage utilization for single channel 8-bit BT.656 display mode. The Y-axis represents percentage utilization and X-axis represents the different FPS for fixed 27 MHz pixel clock.



**Figure 26. Utilization for Single Channel 8-Bit BT.656 Display Mode**

#### 4.3.10 Video Port Configured as Y/C Video Capture Mode

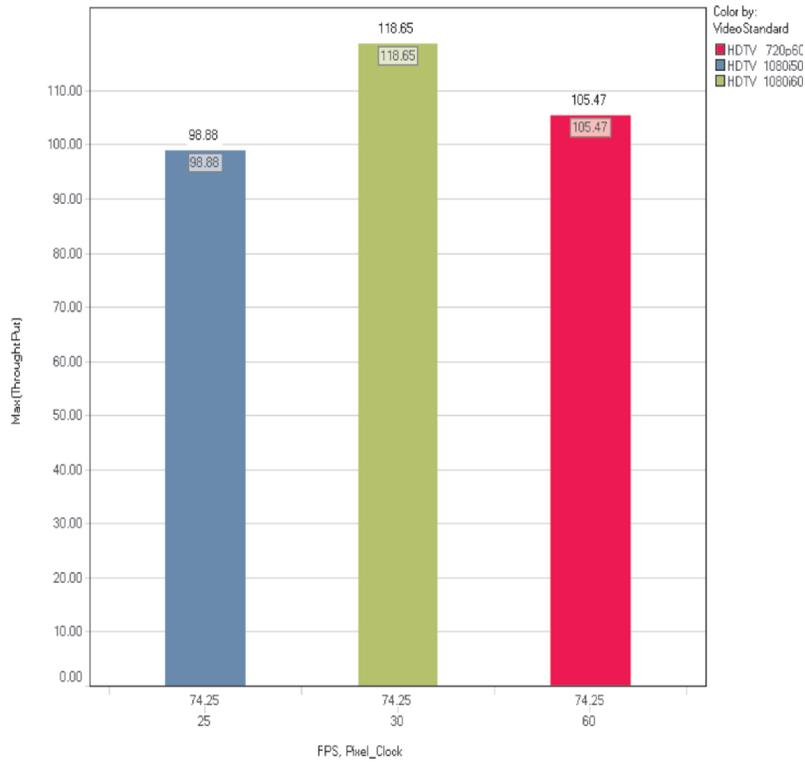
##### 4.3.10.1 Test Environment

System setup for the single channel 8-bit Y/C video capture mode throughput measurement is given below:

- CPU clock: 900 MHz.
- DDR clock: 266 MHz.
- Video Port: Port 0
- EDMA Transfer Controller: TC2
- Video port mode: Single channel 8-bit Y/C video capture
- Throughput data collected is standalone. No other ongoing traffic.
- Throughput data collected using NDK drivers.

**4.3.10.2 Performance of Video Port**

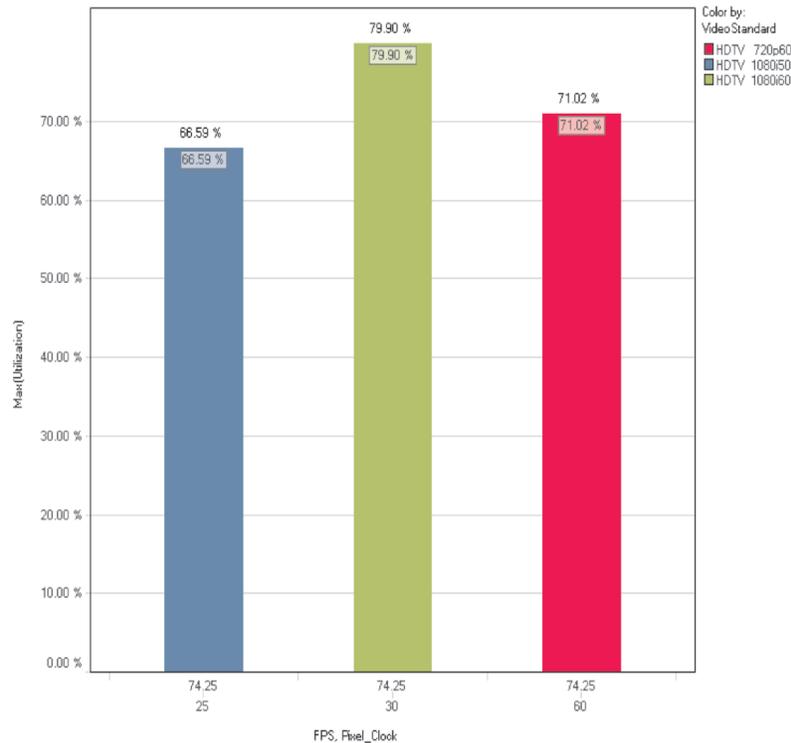
Figure 27 shows the throughput for single channel 8-bit Y/C capture mode. The Y-axis represents the throughput in MB/sec and X-axis represents the different FPS for fixed 74.25 MHz pixel clock.



**Figure 27. Throughput for Single Channel 8-Bit Y/C Capture Mode**

For interlaced scanning, FPS is 25 and 30. For progressive scanning, FPS is 60 and is used in Figure 27.

Figure 28 shows the percentage utilization for single channel 8-bit Y/C capture mode. The Y-axis represents percentage utilization and X-axis represents the different FPS for fixed 74.25 MHz pixel clock.



**Figure 28. Utilization for Single Channel 8-Bit Y/C Capture Mode**

#### 4.3.11 Video Port Configured as Y/C Video Display Mode

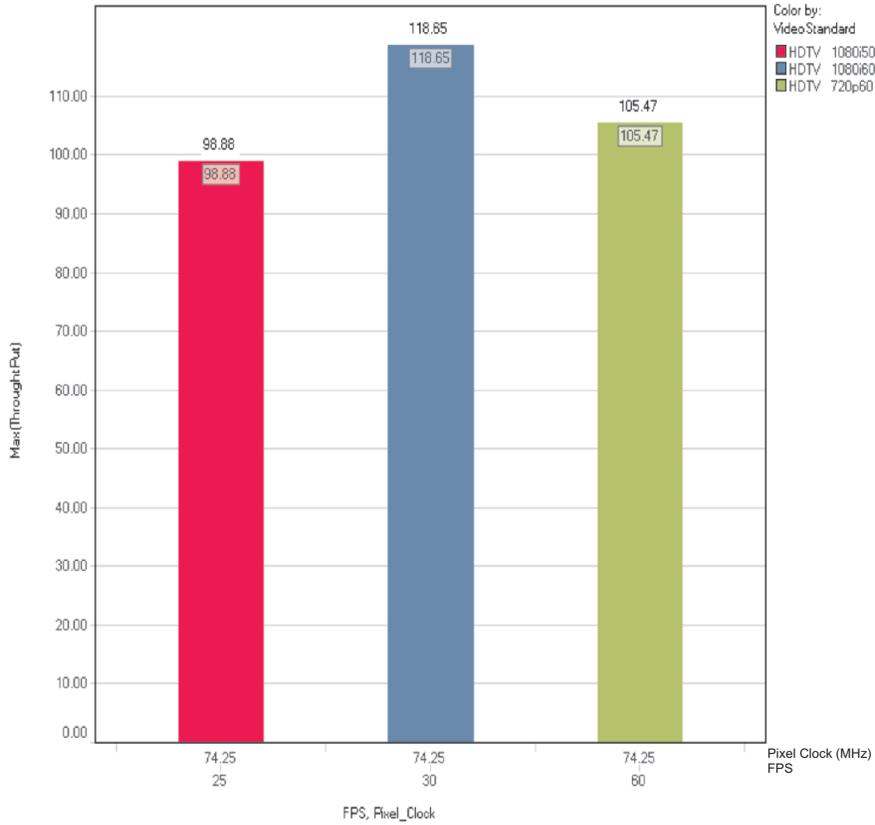
##### 4.3.11.1 Test Environment

System setup for a single channel Y/C 8-bit display mode throughput measurement is given below:

- 1 CPU clock: 900 MHz.
- 1 DDR clock: 266 MHz.
- 1 VideoPort: port 1
- 1 EDMA Transfer Controller: TC3
- 1 Video port mode: Single channel 8-bit Y/C video display
- 1 Throughput data collected is standalone. No other ongoing traffic.
- 1 Throughput data collected is standalone. No other ongoing traffic.

**4.3.11.2 Performance of Video Port**

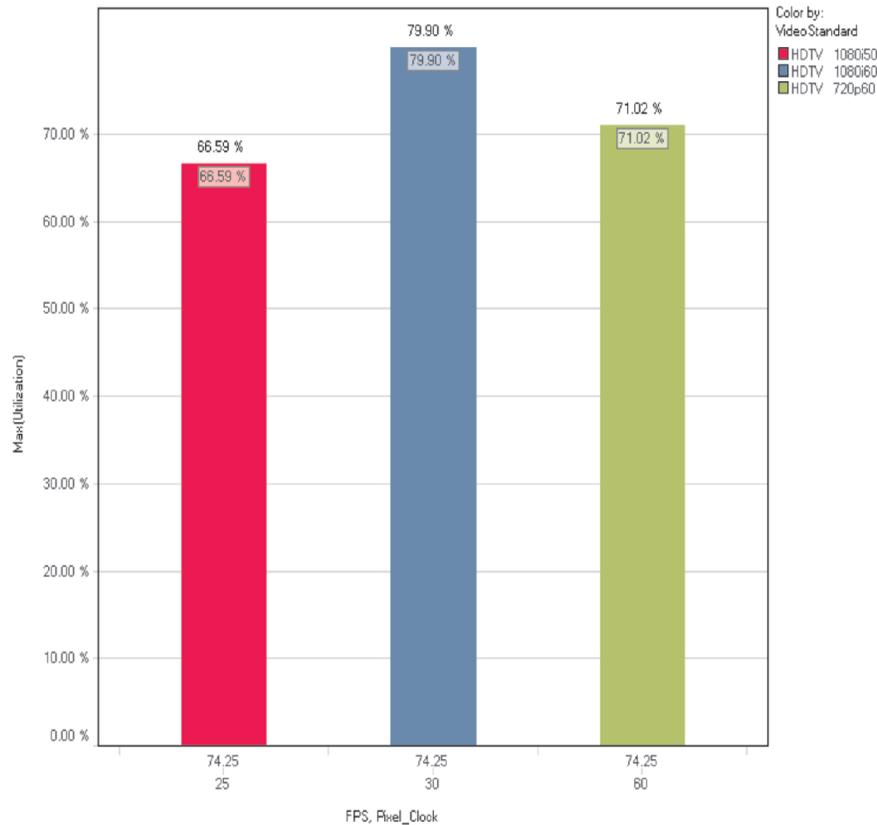
Figure 291 shows the throughput for single channel 8-bit Y/C display mode. The Y-axis represents the throughput in MB/sec and X-axis represents the different FPS for fixed 74.25 MHz pixel clock.



**Figure 29. Throughput for Single Channel 8-Bit Y/C Display Mode**

For interlaced scanning, FPS is 25 and 30. For progressive scanning FPS, is 60 and used in Figure 29.

Figure 30 shows the percentage utilization for single channel 8-bit Y/C display mode. The Y-axis represents percentage utilization and X-axis represents the different FPS for fixed 74.25 MHz pixel clock.



**Figure 30. Utilization for Single Channel 8-Bit Y/C Display Mode**

### 4.3.12 Video Port Configured as Raw Video Display Mode

#### 4.3.12.1 Test Environment

System setup for a single channel 16-bit raw video display mode throughput measurement is given below:

- CPU clock: 900M Hz.
- DDR clock: 266 MHz.
- Video Port: port 1
- EDMA Transfer Controller: TC3
- Video port mode: Single channel 16-bit raw video display
- Throughput data collected is standalone. No other ongoing traffic.
- Throughput data collected using NDK drivers.

4.3.12.2 Performance of Video Port

Figure 31 shows the throughput for single channel 16-bit raw video display mode. The Y-axis represents the throughput in MB/sec and X-axis represents the different FPS and pixel clock.

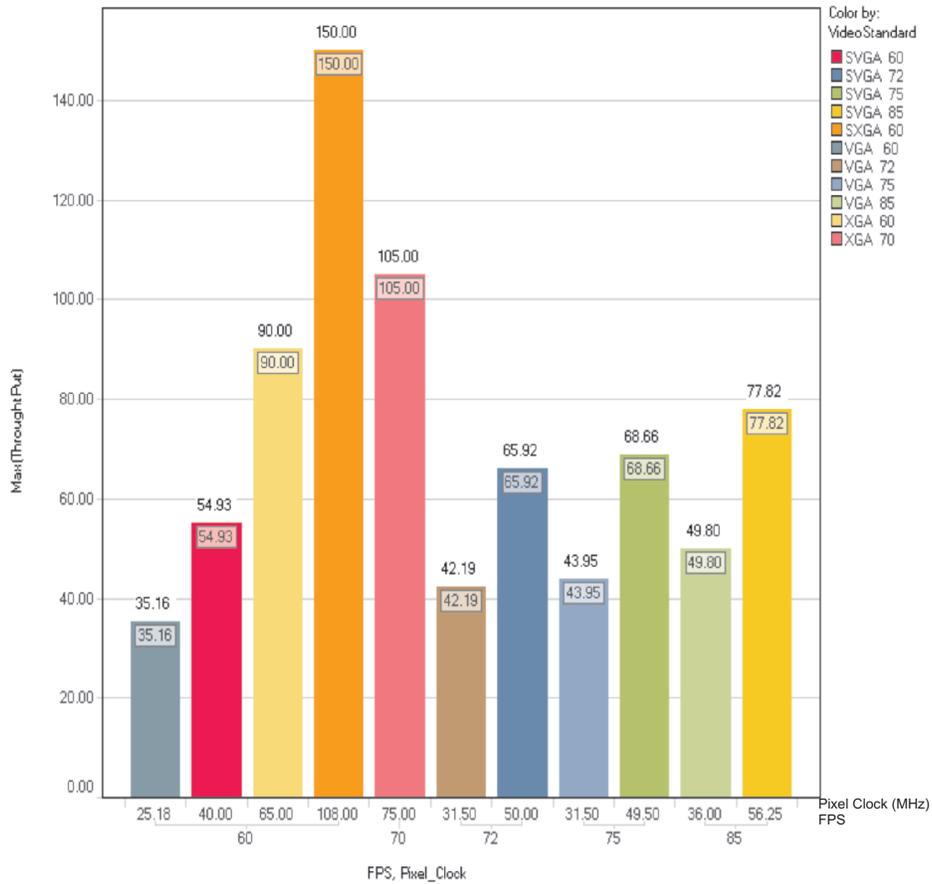
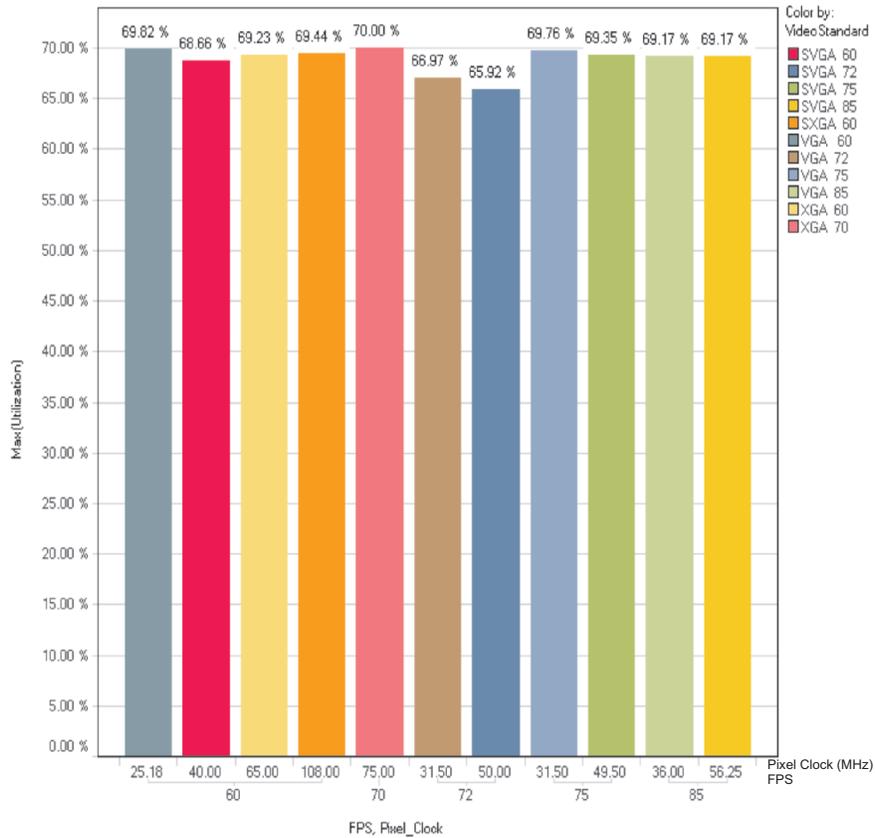


Figure 31. Throughput for Single Channel 16-Bit Raw Video Display Mode

For the fixed FPS, as pixel clock increases throughput also increases and for low pixel clock frequency throughput is very low.

Figure 32 shows the percentage utilization for single channel 16-bit raw video display mode. The Y-axis represents percentage utilization and X-axis represents the different FPS and pixel clock.



**Figure 32. Utilization for Single Channel 16-Bit Raw Video Display Mode**

Utilization of bandwidth almost remains constant irrespective of increase in pixel clock with respect to given FPS.

## 4.4 Ethernet Subsystem

This section provides the throughput analysis of the ESS module integrated in the TMS320DM647/648 SoC.

### 4.4.1 Overview

The Ethernet module controls the flow of packet data between the device and two external Ethernet PHYs (DM648 only) or one external Ethernet PHY (DM647) only, with hardware flow control and quality-of-service (QoS) support. The Ethernet subsystem contains a 3-port gigabit switch, where one port is internally connected to the C64X+ DSP via the switched central resource and the other two ports are brought out externally. It provides the serial gigabit media independent interface (SGMII) and the management data input output (MDIO) for physical layer (PHY) device management.

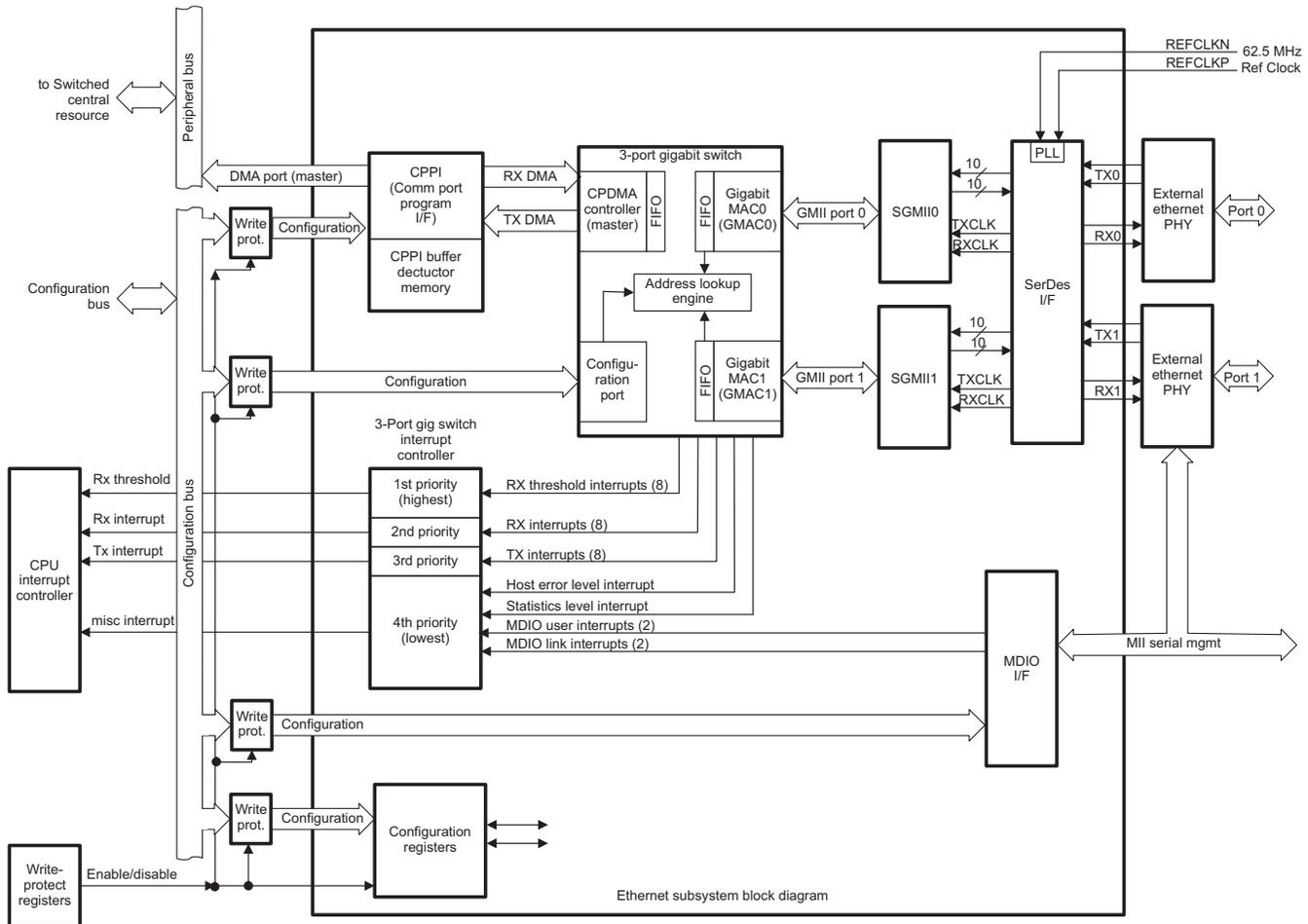


Figure 33. 3PSW Block Diagram

3PSW Ethernet subsystem clock is derived from REFCLKP/N and SerDes PLL programming. REFCLKP/N should be in the range of 50 MHz to 62.5 MHz.

#### 4.4.1.1 3PSW

The 3PSW block contains the following functions:

- The CPDMA sub module is a CPPI 3.0 complaint packet DMA transfer controller. Host software sends and receives network frames via the CPPI3.0 complaint Host interface. The Host interface includes module registers and Host memory data structures. The Host memory data structures are CPPI3.0 buffer descriptors and data buffers. Buffer descriptors may be linked together to describe frames or queues of frames for data transmission and free buffer queues that are available for received data.
- GMAC performs synchronous 10/100/1000 Mbit operation, provides gigabit media independent interface (GMII). Handles hardware error handling including CRC, it operates in full duplex gigabit, provides etherStats and 802.3 Stats RMON statistics gathering support for external statistics collection module, provides emulation support, supports VLAN aware mode
- The address lookup engine (ALE) processes all received packets to determine where to forward the packet. The ALE uses the incoming packet received port number, destination address, source address, length/type, and VLAN information to determine how the packet should be forwarded.

#### 4.4.1.2 Serial Gigabit Media Independent Interface (SGMII)

The SGMII receives RX interface, then converts the encoded receive input from the SerDes into the required GMAC signals. The SGMII transmit TX interface converts the GMAC GMII input data in to the required encoded transmit outputs. It does 8B/10B encoding and decoding and serializer/deserializer technology.

#### 4.4.1.3 Serializer/Deserializer (SerDes) Module

The SerDes converts parallel data to serial data and vice-versa. The transmitter section is parallel-to-serial converter, and the receive section is serial-to-parallel converter.

#### 4.4.1.4 MDIO

The MII management I/F module implements the 802.3 serial management interfaces to interrogate and control two Ethernet PHYs simultaneously using a shared two-wire bus. Prior to initiating any other transaction, the station management entity sends a preamble sequence of 32 contiguous logic one bits on the MDIO line with 32 corresponding cycles on MDCLK to provide the PHY with a pattern that it can use to establish synchronization. A PHY observes a sequence of 32 contiguous logic one bits on MDIO with 32 corresponding MDCLK cycles before it responds to any other transaction

#### 4.4.1.5 TX Operation

Word 0

31		0
NEXT_DESCRIPTOR_POINTER		

Word 1

31		0
BUFFER_POINTER		

Word 2

31		16	15		0
BUFFER_OFFSET			BUFFER_LENGTH		

Word 3

31	30	29	28	27	26	25	24	
SOP	EOP	OWNERSHIP	EOQ	TEARDOWN_COMPLETE	PASS_CRC	Reserved		
23		21		20		18	17	16
Reserved			TP_PORT_EN			Reserved		TO_PORT
15		11	10					0
Reserved				PACKET_LENGTH				

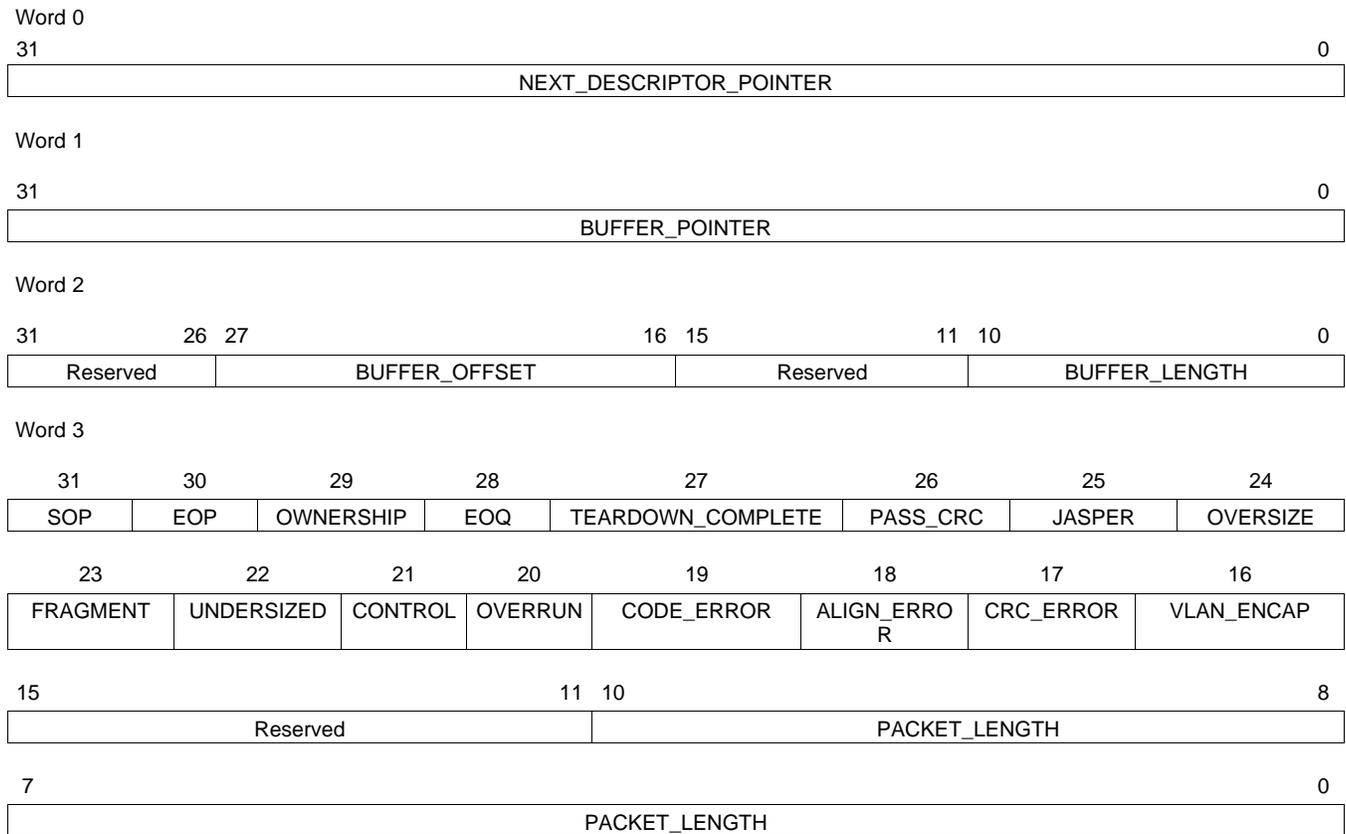
**Figure 34. Transmit Buffer Descriptor**

After reset, the Host must write zeros to all transmit DMA state head descriptor pointers. To initiate packet transmission, the Host constructs transmit queues in the memory (one or more packets for transmission) and then writes the appropriate transmit DMA state head descriptor pointers. The port begins TX packet transmission on a given channel when the Host writes the channels transmit queue head descriptor pointer with the address of the first buffer descriptor in the queue (nonzero value). The first buffer descriptor for each transmit packet must have the SOP bit and the OWNERSHIP bit set to one by the Host. The last buffer descriptor for each transmit packet must have the EOP bit set to one by the Host. The port will transmit packets until all queued packets have been transmitted and the queue(s) are empty.

When each packet transmission is complete, the port clears the OWNERSHIP bit in the packet's SOP buffer descriptor and issues an interrupt to the Host by writing the packets last buffer descriptor address to the queues transmit DMA state completion pointer. When the last packet in the queue has been transmitted, the port sets the EOQ bit in EOP buffer descriptor, and clears the OWNERSHIP bit in the SOP descriptor.

On the interrupt from the port, the Host processes the buffer queue detecting transmitted packets by the status of the OWNERSHIP bit in SOP buffer descriptor. If the OWNERSHIP bit is cleared, then the packet has been transmitted and the Host can reclaim the buffers associated with the packet. The Host continues queue processing until an SOP buffer descriptor is read that contains a set OWNERSHIP bit indicating that the transmission is not complete. The Host determines that all packets in the queue have been transmitted when the OWNERSHIP bit in the EOP buffer descriptor is cleared, the EOQ bit is set in the last packet EOP buffer descriptor, and the NEXT\_DESCRIPTOR\_POINTOR of the last packet EOP buffer descriptor is set to zero. The Host acknowledges an interrupt by writing the address of the last buffer descriptor to the queues associated transmit completion pointer in the transmit DMA state. If the Host written buffer address value is different from the buffer address written by the port, then the interrupt remains asserted. If it matches, then the interrupt is de-asserted.

**4.4.1.6 RX Operation**



**Figure 35. Receive Buffer Descriptor**

After reset, the Host must write zeros to all receive DMA state head descriptor pointers. To initiate packet reception, the Host constructs receive queues in the memory (one or more packets for transmission) and then writes the appropriate transmit DMA state head descriptor pointers. The Host enables packet reception on a given channel by writing the address of the first buffer descriptor in the queue (nonzero value) to the channels head descriptor pointer in the channel receive DMA state. When packet reception begins on a given channel, the port fills each receive buffer with data in the order starting with the first buffer and proceeding through the receive queue. At the end of each packet reception, the port overwrites the BUFFER\_LENGTH in the packets EOP buffer descriptor with the number of bytes actually received in the packets last buffer, sets the EOP bit in the packets EOP buffer descriptor, sets the EOQ bit in the EOP buffer descriptor when the last packet is in queue, sets the SOP bit in the packets SOP buffer descriptor, writes the SOP buffer descriptor PACKET\_LENGTH field, clears the OWNERSHIP bit in the packets SOP buffer descriptor, and issues the receive Host interrupt by writing the address of the packets last buffer descriptor to the queues receive DMA state completion pointer.

Upon interrupt, if the OWNERSHIP bit is cleared, then the packet has been received completely and is available to be processed by the Host. But if the OWNERSHIP bit is not cleared, the Host will continue to receive queue processing until the end of the queue. The Host determines that the queue is empty when the last packet in the queue has cleared the OWNERSHIP bit in the SOP buffer descriptor and EOQ is set in the EOP buffer descriptor. The NEXT\_DESCRIPTOR\_POINTER in the EOP buffer descriptor is zero.

#### 4.4.2 Factors Affecting ESS Throughput Value

To gain high throughput and the best performance, ESS has to be configured properly.

The different factors considered for throughput calculation with its impact is given in [Table 16](#)

**Table 16. Factors considered for throughput**

Factors	Impact	Recommendation
Rate Scale	When rate scale is full, performance is good. When rate scale is half or quarter line, rate also decreases.  Note: Refer to the PRG for valid rate, REFCLK combinations, MPY(multiplication factor).	
Line rate (10/100/1000)	Performance degrades when data rate decreases.	Choose the highest possible line rate.
Packet Size	Performance is low for smaller packet size.	Configure for maximum packet size specified in the data sheet.

The video port includes a FIFO to store data coming into or out of the video port. The video port operates in conjunction with EDMA transfers to move data between the video port FIFO and external or on-chip memory. EDMA events are generated when the video port FIFO reaches certain fullness (for capture) or goes below certain fullness (for display).

#### 4.4.3 Performance of ESS

[Table 17](#) captures the throughput and percentage utilization for different ESS.

**Table 17. ESS Performances**

Protocol type	Ethernet Frame Size in Bytes	Bandwidth Mbits/Sec	Ideal Throughput in Mbits/Sec	Utilization
ICMP	64	55	1000	5.50%
ICMP	128	95.6	1000	9.56%
ICMP	256	171	1000	17.10%
ICMP	512	320	1000	32.00%
ICMP	1024	600	1000	60.00%
ICMP	1518	601	1000	60.10%
TCP	64	49	1000	4.90%

**Table 17. ESS Performances (continued)**

Protocol type	Ethernet Frame Size in Bytes	Bandwidth Mbites/Sec	Ideal Throughput in Mbites/Sec	Utilization
TCP	128	85	1000	8.50%
TCP	256	155	1000	15.50%
TCP	512	286	1000	28.60%
TCP	1024	524	1000	52.40%
TCP	1518	716	1000	71.60%

#### 4.4.4 ESS ICMP tThroughput Measurement

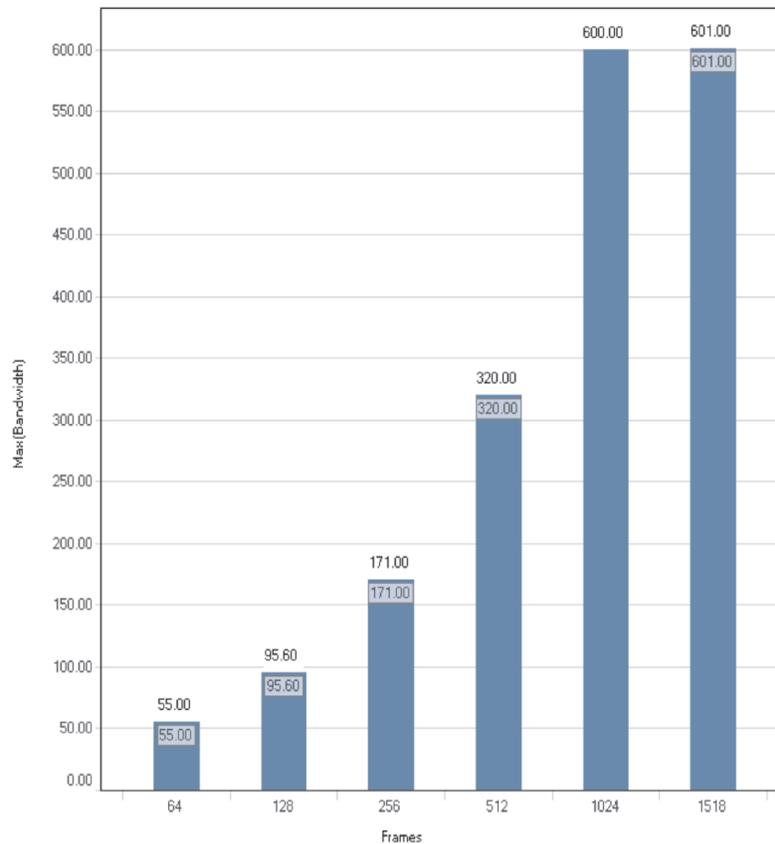
##### 4.4.4.1 Test Environment

System setup for ESS with ICMP protocol throughput measurement is given below:

- CPU frequency: 900 MHz
- DDR2 frequency: 266 MHz
- ESS frequency: 62.5 MHz
- ESS ports used:
- Throughput data collected is standalone. No other ongoing traffic.
- Throughput data collected using NDK drivers.

#### 4.4.4.2 Performance of ESS

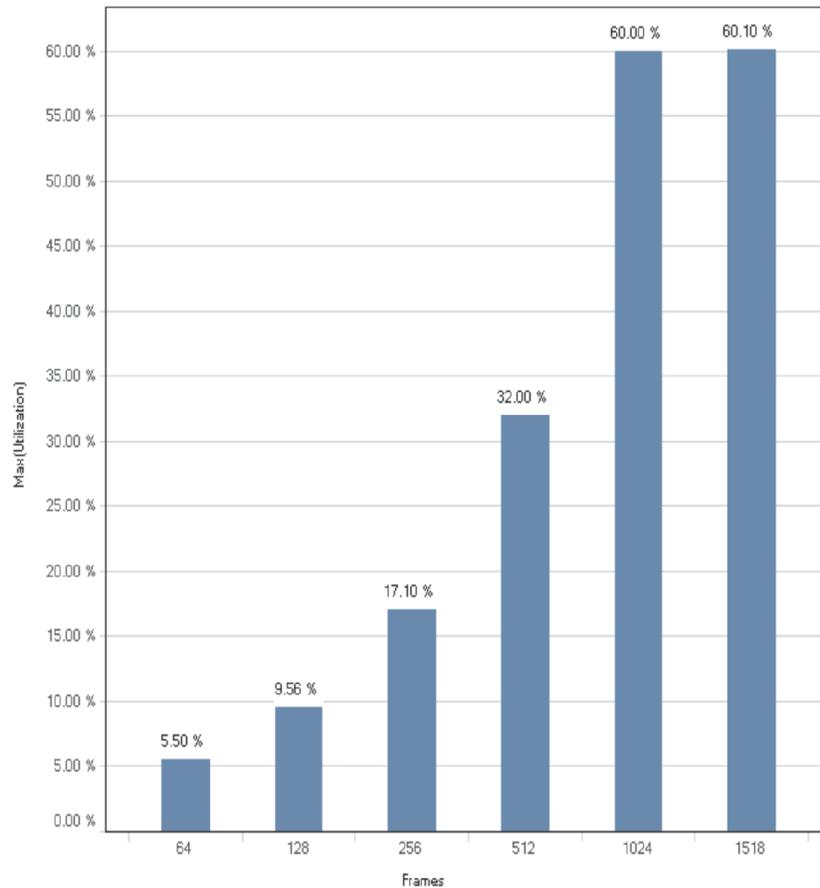
Figure 36 shows the throughput for ESS with ICMP protocol. Y-axis represents the throughput in MB/Sec and X-axis represents the Ethernet frame size.



**Figure 36. Throughput for ESS With ICMP Protocol**

As frame size increased, throughput also increased and for frame size 1518 bytes, actual throughput 601 MB/Sec is observed. For frame size above 1518 bytes and throughput above 601 MB/Sec, packet loss is observed. So maximum actual throughput observed is 601 MB/Sec.

Figure 37 shows the percentage of utilization for ESS with ICMP protocol. The Y-axis represents percentage of utilization and X-axis represents the Ethernet frame size.



**Figure 37. Percentage of Utilization of ESS With ICMP**

Performance increases as Ethernet frame size increases. And maximum performance seen is 60.10% for Ethernet frame size 1518 bytes.

#### 4.4.5 ESS TCP throughput measurement

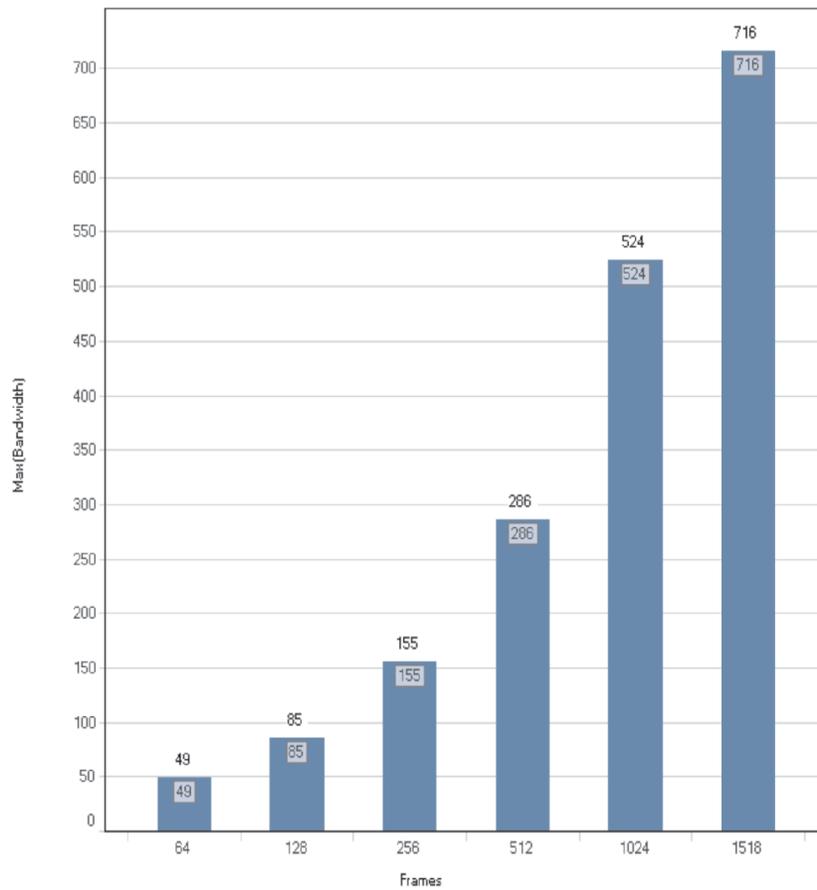
##### 4.4.5.1 Test Environment

System setup for ESS with TCP protocol throughput measurement is given below:

- CPU frequency: 900 MHz
- DDR2 frequency: 266 MHz
- ESS frequency: 62.5 MHz
- ESS ports used:
- Throughput data collected is standalone. No other ongoing traffic.
- Throughput data collected using NDK drivers.

#### 4.4.5.2 Performance of ESS

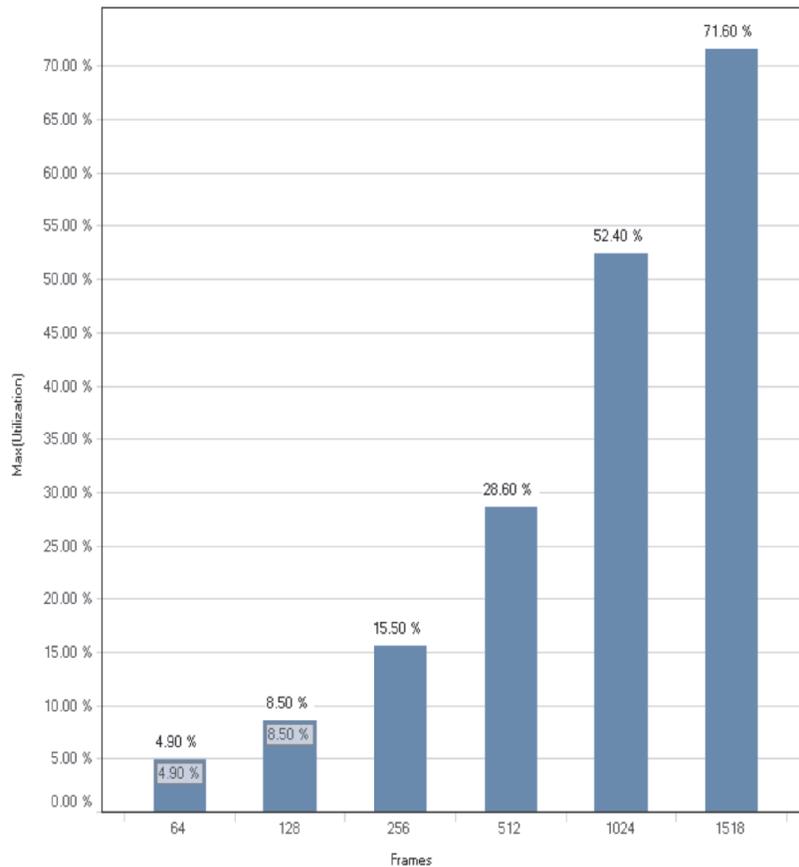
Figure 38 shows the throughput for ESS with TCP protocol. Y-axis represents the throughput in MB/Sec and X-axis represents the Ethernet frame size.



**Figure 38. Throughput for ESS With TCP Protocol**

As frame size increased, throughput also increased; for frame size 1518 bytes, actual throughput 716 MB/Sec is observed. For frame size above 1518 bytes and throughput above 716 MB/Sec, packet loss is observed. So maximum actual throughput observed is 716 MB/Sec.

Figure 39 shows the percentage of utilization for ESS with TCP protocol. The Y-axis represents percentage of utilization and X-axis represents the Ethernet frame size.



**Figure 39. Percentage of Utilization of ESS With TCP**

Performance increases as Ethernet frame size increases. And maximum performance seen is 71.60% for Ethernet frame size 1518 bytes.

## 5 References

- *TMS320DM647/TMS320DM648 Digital Media Processors* ([SPRS372](#))
- *TMS320DM647/DM648 DSP DDR2 Memory Controller User's Guide* ([SPRUEK5](#))
- *TMS320DM647/DM648 DSP Multichannel Audio Serial Port (McASP) User's Guide* ([SPRUEL1](#))
- *TMS320DM647/DM648 DSP Enhanced DMA (EDMA3) Controller User's Guide* ([SPRUEL2](#))
- *TMS320DM647/DM648 Video Port/VCXO Interpolated Control (VIC) Port User's Guide* ([SPRUEM1](#))

## 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	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
RF/IF and ZigBee® Solutions	<a href="http://www.ti.com/lprf">www.ti.com/lprf</a>

### Applications

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Broadband	<a href="http://www.ti.com/broadband">www.ti.com/broadband</a>
Digital Control	<a href="http://www.ti.com/digitalcontrol">www.ti.com/digitalcontrol</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Military	<a href="http://www.ti.com/military">www.ti.com/military</a>
Optical Networking	<a href="http://www.ti.com/opticalnetwork">www.ti.com/opticalnetwork</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Telephony	<a href="http://www.ti.com/telephony">www.ti.com/telephony</a>
Video & Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
Wireless	<a href="http://www.ti.com/wireless">www.ti.com/wireless</a>

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