Application Report
SPRABK5B–November 2011–Revised December 2015

Throughput Performance Guide for KeyStone II Devices

Multicore Processors

ABSTRACT

This application report analyzes various performance measurements of the KeyStone II family of processors. It provides a throughput analysis of the various support peripherals to different end-points and memory access.

Contents

List of Figures

List of Tables

1 Introduction

The purpose of this document is to provide throughput performance data for Keystone II Architecture devices. This document provides theoretical and measured throughput performance for Keystone II memories and peripherals. Note that not all peripherals are supported by all devices. For a particular device, see the table or the device-specific data manual to determine which peripherals are supported.

In addition to throughput performance data, this document also provides an overview of the TeraNet switch fabric implemented on Keystone II devices. Lastly, this document provides a basis for estimating memory access performance and presents theoretical and measured performance data achieved under various operating conditions. Some factors affecting memory access performance also are detailed.

1.1 Acronyms Used In This Document

Acronym	Definition
DSP	Digital Signal Processor
DDR	Dual Data Rate
EMIF	External Memory Interface Controller
EDMA3	Enhanced Direct Memory Access v3.0
LSU	Load Store Unit
MSMC	Multicore Shared Memory Controller
PCIe	Peripheral Component Interconnect Express
SDMA	Slave Direct Memory Access
SDRAM	Synchronous Dynamic Random Access Memory
SRIO	Serial Rapid Input Output
TPDMA	Third Party DMA Engine
TPCC	TPDMA Channel Controller
TPTC	TPDMA Transfer Controller
XMC	Extended Memory Controller

2 KeyStone II Device Overview

This section focuses on the TeraNet switch fabric that provides the interconnect between C66x CorePac, ARM CorePac, peripherals, and memories. The high-level details provided in this section are required to understand the throughput performance of the C66x DSP, because the bus widths and the operating frequencies of each part of the TeraNet directly impact the throughput performance of the connected peripherals or memories.

2.1 C66x CorePac Overview

This section provides the fundamental details of the C66x CorePac. Additional details about the C66x CorePac can be found in the *C66x CorePac User's Guide* (SPRUGW0). The C66x CorePac consists of several components including:

- C66x DSP core
- L1 and L2 memories
- External memory controller
- Extended memory controller (XMC)
- Interrupt controller
- Power and sleep controller
- Embedded trace buffer
- Data trace formatter
- RSA accelerator (core 1 and 2 only)

Each CorePac has the ability to sustain up to 128 bits of load/store operations per cycle to L1D memory and is capable of handling up to 16 GB/second. When accessing data in the L2 memory or external memory, the access rate depends on the memory access pattern and the cache. The cache size varies depending on the platform.

Within each CorePac, there is an internal DMA (IDMA) engine that can move data at the frequency rate of DSP/2 with a data width of 256 bits. The IDMA operates in the background of the DSP core activity (i.e., data can be brought into buffer A while the DSP core is accessing buffer B). The IDMA can transfer data only between the L1 and L2 memories and peripheral configuration ports; it cannot access external memory.

2.2 ARM CorePac Overview

This section provides the fundamental details of the ARM CorePac. Additional details about the ARM CorePac can be found in the *ARM CorePac User's Guide* (SPRUGW0). The ARM CorePac consists of several components including:

- Cortex -A15 processor revision R2P4
- L1 and L2 cache memories
- NEON Media Coprocessor
- Interrupt Controller
- AXI protocol external interface
- Non-invasive Debug Support
- Clock Generation (dedicated ARM PLL)

Each ARM CorePac is capable of supporting 16 word line access using a 128-bit interface to L1 and L2 instruction and data cache. The KeyStone II ARM CorePac supports a shared L2 cache across the cluster and the cache size varies depending on the number of ARM cores (4MB quad, 1MB dual and 512KB). The NEON coprocessor implements the advanced SIMDv2 media processing architecture and the VFPv4 Floating Point architecture. The external interface uses the AXI protocol configured to 128-bit data width.

Within each CorePac, there is an internal DMA (IDMA) engine that can move data at the frequency rate of DSP/2 with a data width of 256 bits. The IDMA operates in the background of the DSP core activity (i.e., data can be brought into buffer A while the DSP core is accessing buffer B). The IDMA can transfer data only between the L1 and L2 memories and peripheral configuration ports; it cannot access external memory.

2.3 TeraNet Overview

The TeraNet switch fabric provides interconnection between the C66x CorePac (and their local memories), MSMC memory, ARM CorePac, external memory, the enhanced DMA v3 (EDMA3) controllers, Multicore Navigator, on-chip coprocessors, and high-speed IO. The TeraNet switch fabric allows each of these to operate at maximum efficiency with no blocking or stalling. It allows for concurrent transfers between non-conflicting master/slave pairs and can support a very high total data rate across any endpoint. If transfers line up such that the source or destination memory is the same, then collisions occur and certain transactions will be blocked.

The TeraNet consists of data switch fabric and configuration switch fabric:

- **Data Switch Fabric:** The data switch fabric mainly moves data across the system and is further subdivided into two smaller switch fabrics. One connects high speed masters to slaves via 256-bit data buses running at the DSP/3 frequency. The second data switch fabric connects high-speed masters to slaves via 128-bit data buses running at a DSP/3 frequency.
- **Configuration Switch Fabric:** This switch fabric is mainly used to access peripheral registers. It connects the C66x CorePac and masters on the data switch fabric to slaves via 32-bit configuration buses running at either DSP/3 or DSP/6 frequency.

KeyStone II devices contain up to five EDMA Channel Controllers: TPCC0, TPCC1, TPCC2, TPCC3, and, TPCC4. Each TPCC can be programmed to move data concurrently in the background without the expense of any DSP cycles. These TPCCs can move data between on-chip L1 and L2 memory, MSMC SRAM, external memory, and the peripherals on the device that support EDMA-based transfer.

The TeraNet also provides connection to the Multicore Navigator which uses a Queue Manager Subsystem (QMSS) and a Packet DMA (PKTDMA) to control and implement high-speed data packet movement within the device. Frequent tasks are commonly off-loaded from the host processor to peripheral hardware to increase system performance.

The TeraNet also provides a 64-bit DDR3A and 64-bit DDR3B interface to support access to external memories that can be used for either data or program memory.

2.4 Memory Access System Overview

This section discusses the KeyStone II memory system. Memory access is critical for applications running on the device. Some memories are internal to the CorePacs, while other system memories are external to the CorePacs. The memory system also provides a 64-bit DDR3A and 64-bit DDR3B interface for accessing off-chip memory.

Internal to each DSP CorePac are the following memories:

- DSP L1D SRAM that can be used as either data memory or data cache or both.
- DSP L1P SRAM that can be used as either program memory or cache or both.
- DSP Local L2 SRAM that can be used as either unified SRAM or unified cache or both.

Internal to each ARM CorePac are the following memories:

- ARM L1D for data cache.
- ARM L1P for program memory cache.
- Shared and coherent ARM L2 cache for both data and program memory.

DSP L1D SRAM, DSP L1P SRAM, DSP L2 SRAM, MSMC SRAM, and DDR3 memories are accessible by all the cores and multiple DMA masters on the device.

External to the CorePacs, the KeyStone II devices have multicore shared memory (MSMC) SRAM, which is shared between the cores. The MSMC memory can be configured in two ways:

- **SL2 mode**: Shared L2 SRAM mode: L1P/L1D memory will cache MSMC, whereas L2 will not cache requests to MSMC SRAM. Only available for C66x DSP and not for ARM A15.
- **SL3 mode**: Level 3 SRAM mode: Both L1P/L1D and L2 memories will cache the MSMC SRAM if it is remapped to an external address using the address extension unit. For more information, see the *KeyStone Architecture Multicore Shared Memory Controller (MSMC) User's Guide* (SPRUGW7)

A 64-bit EMIF interface is provided for accessing off-chip DDR3A and DDR3B SDRAM, which can be used as data or program memory. Although this interface supports a 64-bit data bus, it can also be configured to operate using a 32-bit or 16-bit data bus.

3 Memory Access Throughput Performance

This section discusses the memory access throughput performance of KeyStone II devices. The bandwidth of a memory copy is determined by the lowest of these three factors:

- Bus bandwidth
- Source throughput
- Destination throughput

Table 1 summarizes the maximum theoretical bandwidth of the C66x core, IDMA, and the EDMA when the Main PLL is operating at 1.2 GHz.

Table 1. Theoretical Bandwidth of Core, IDMA and EDMA

Master	Maximum Bandwidth MB/s	Comments
C66x Core	19200	(128bits)/(8bit/byte)*(1200M)=19200MB/s
ARM A15 Core	19200	(256bits)/(8bit/byte)*(1200M/2)=19200MB/s
EDMA0(Single TC)	19200	(256bits)/(8bit/byte)*(1200M/2)=19200MB/s [1]
EDMA1(Single TC)	6400	(128bits)/(8bit/byte)*(1200M/3)=6400MB/s
EDMA2(Single TC)	6400	(128bits)/(8bit/byte)*(1200M/3)=6400MB/s
EDMA3(Single TC)	6400	(128bits)/(8bit/byte)*(1200M/3)=6400MB/s
EDMA4(Single TC)	19200	(256bits)/(8bit/byte)*(1200M/2)=12800MB/s [2]

[1] 66AK2E and AM5K2E devices have EDMA_0 at CPU/3 on the TeraNet CPU/3 node.

[2] For AM5K2E and 66AK2E devices, EDMA0 is at CPU/3 speed resulting in a theoretical bandwidth of (256 bits)/(8 bit/byte)*(1200M/3)=12800MB/s.

Table 2 summarizes the maximum theoretical throughput of different memories when the C66x Main PLL is operating at 1.2 GHz. The DDR3 performance assumes that a 64-bit bus width is used and that the external memory is operating at 1600 Mtps.

Table 2. Theoretical Bandwidth of Different Memories

Master	Maximum Bandwidth MB/s	Comments
DSP L1D	38400	(256bits)/(8bit/byte)*(1200M)=38400MB/s
DSP L1P	38400	(256bits)/(8bit/byte)*(1200M)=38400MB/s
DSP L2	16000	(256 bits)/(8bit/byte)*(1200M/2)=16000MB/s
MSMC RAM	38400	(256bits)/(8bit/byte)*(1200M)=38400MB/s
DDR3A RAM	12800	(64 bits)/(8bit/byte)*(800M)*2=12800MB/s
DDR3B RAM	12800	(64 bits)/(8bit/byte)*(800M)*2=12800MB/s

When the C66x core tries to read from or write to different memory endpoints, it consumes some DSP cycles to access the memory region depending on various factors such as prefetching, caching, victim buffer hits, and so on. Section 3.1 and Section 3.2 estimate the number of DSP stalls for accessing different memory endpoints.

Note that MSMC SRAM has up to 8 memory banks of 256 bits. As mentioned previously, MSMC can be configured in either shared L2 (SL2) mode or shared L3 (SL3) mode. Also, the latencies to ARM CorePac L2 memory and DDR3 memory varies based on the number of ARM cores in the SoC.

3.1 Memory Read Performance

The C66x core has an improved pipeline between L1D/L1P and L2 memory controller and this significantly reduces the stall cycles for L1D/L1P cache misses.

Table 3 shows the comparison of KeyStone I to KeyStone II DSP stalls for accessing different memories when the DSP core tries to read from memory.

Table 3. C66x DSP Memory Read Performance

	KeyStone I		KeyStone II	
	DSP Stalls in cycles		DSP Stalls in cycles	
Source	**Single Read**	**Burst Read**	**Single Read**	**Burst Read**
L1D Hit	0	0	0	0
L2 SRAM	7	3.5	7	3.5
L2 Cache Hit	9	5	9	5
MSMC RAM (SL2)	20	7	22	7
MSMC RAM (SL3)	23	10	25	8
DDR 3A RAM (SL3)	100	31	103	30
DDR 3B RAM (SL3)	N/A	N/A	152	43

The test case details are as follows:

- KeyStone I testing is done on a 1.2 GHz DSP with MSMC at 600MHz and DDR frequency at 1600 MT/s.
- KeyStone II testing is done on a 1.2GHz DSP with MSMC at 1.2GHz and DDR3 frequency at 1600 MT/s.
- Single read scenarios represent endpoint access latency.
- Burst read scenarios
  - 1 read miss per cycle
  - L2 cache is enabled, cold cache
  - Stride = 128 bits
  - Prefetch is enabled (SL2/SL3)

- No victims assumed for read latency measurements.
- DDR3 accesses to open pages.

Table 4 shows the KeyStone II ARM stalls for accessing different memories when the ARM core tries to read from memory.

Table 4. ARM A15 Memory Read Performance

	KeyStone II	
	A15 load to use latency	
Source	**Single Read**	**Burst Read**
L1 Cache Hit	4	0.8
L2 Cache Hit	20	4.4
MSMC RAM (SL3)	54	12
DDR 3A RAM (SL3)	140	18
DDR 3B RAM (SL3)	197	23

The test case details are as follows:

- KeyStone II testing is done on a 1.4GHz ARM with MSMC at 1.2GHz and DDR3 frequency at 1600 MT/s.
- Single read scenarios represent endpoint access latency.
- Read latency numbers quoted with ARM default cache performance settings.
- DDR3 accesses to open pages.

3.2 Memory Write Performance

The C66x CorePac has improved write merging and optimized burst sizes that reduce the stalls to external memory.

The L1D memory controller merges writes, not only to L2 SRAM, but to any address that is allowed to be cached (MAR.PC==1).

For details about memory write performance, see the *TMS320C66x DSP Cache User's Guide* (SPRUGY8).

4 DDR3 Throughput

Applies to:

All KeyStone II Devices

The DDR3 module in the C66x DSP is accessible across all cores and other system masters. The DDR3 controller interfaces with most standard DDR3 SDRAM devices. There are two clock domains in the controller. The Command FIFO, Read FIFO and Write FIFO are all on the DSP/2 clock domain for DDR3A and are all on DSP/3 clock domain for DDR3B. The state machine and interface to the DDR PHY are all driven by the DDR3 memory clock. The DDR3 controller is interfaced to the MSMC controller and any access made via any of the EDMAs are handled by MSMC that serves as the common management path.

DDR3 throughput can be measured using a bidirectional EDMA transfer setup with DDR3 and other memory endpoints. The theoretical DDR3 throughput is (64bits)/(8bit/byte)×(800M)×2=12800MB/s, it is impacted by factors such as read-write turnaround and page switch overheads'. For DDR3 related throughput, see the Section 5.2.

5 EDMA Throughput

The enhanced direct memory access (EDMA3) controller's primary purpose is to service data transfers between two memory-mapped slave endpoints on the device. An EDMA3 controller consists of the channel controller (EDMA3CC) which serves as the user interface allowing software to program the data transfer and for each channel controller, a number of transfer controllers (TCs) that are responsible for actual data movement. A single transfer controller may not always be able to utilize the desired slave memory bandwidth. For example: With Main PLL running at 1.2GHz a, 128-bit wide TC running at CPU/3 is limited to 128/8 * 1200/3 = 6400 MB/s. 80% of DDR3 bandwidth with 64-bit DDR running at 1600MT/s is 0.8 * (64/8)*1600 = 10240 MB/s. Thus, more than one TC will need to be employed to hit 80% bandwidth.

The EDMA3 throughput section is divided into three subsections:

Section 5.1 provides the baseline reference for EDMA TC throughput. These tests use individual TCs with large, single-dimensional transfers (128KB).

Section 5.2 describes the multi-TC complex throughput tests. The intent is to employ multiple EDMA3 TCs to maximize the end-to-end bandwidth for a given set of slave memory endpoints. These tests utilize large, single-dimensional transfers (32KB) designed to optimize pipelined bus utilization and minimize overhead associated with setting up transfers.

Section 5.3 describes the impact of various parameters like data burst size (DBS), FIFO size and transfer index on the EDMATC throughput. A single transfer controller is used to clearly demonstrate this impact.

All EDMA3 throughput tests were run on a 66AK2H EVM with the Main PLL running at 1.2GHz and 64-bit DDR3 configured at 1600 MT/s.

Note that TC0 through TC13 in this document correlate to the Data Manual nomenclature in the following way:

TC0 = EDMA0 TC0	TC7 = EDMA2 TC1
TC1 = EDMA0 TC1	TC8 = EDMA2 TC2
TC2 = EDMA1 TC0	TC9 = EDMA2 TC3
TC3 = EDMA1 TC1	TC10 = EDMA3 TC0
TC4 = EDMA1 TC2	TC11 = EDMA3 TC1
TC5 = EDMA1 TC3	TC12 = EDMA4 TC0
TC6 = EDMA2 TC0	TC13 = EDMA4 TC1

5.1 *Baseline EDMA TC Performance*

This section compares the maximum throughput achievable for TCs on various channel controllers found on Keystone II devices. EDMA was programmed in AB-Sync mode with ACNT=1024, BCNT=128, CCNT=1.

NOTE: Some small variations in the results shown may be attributed to measurement accuracy.

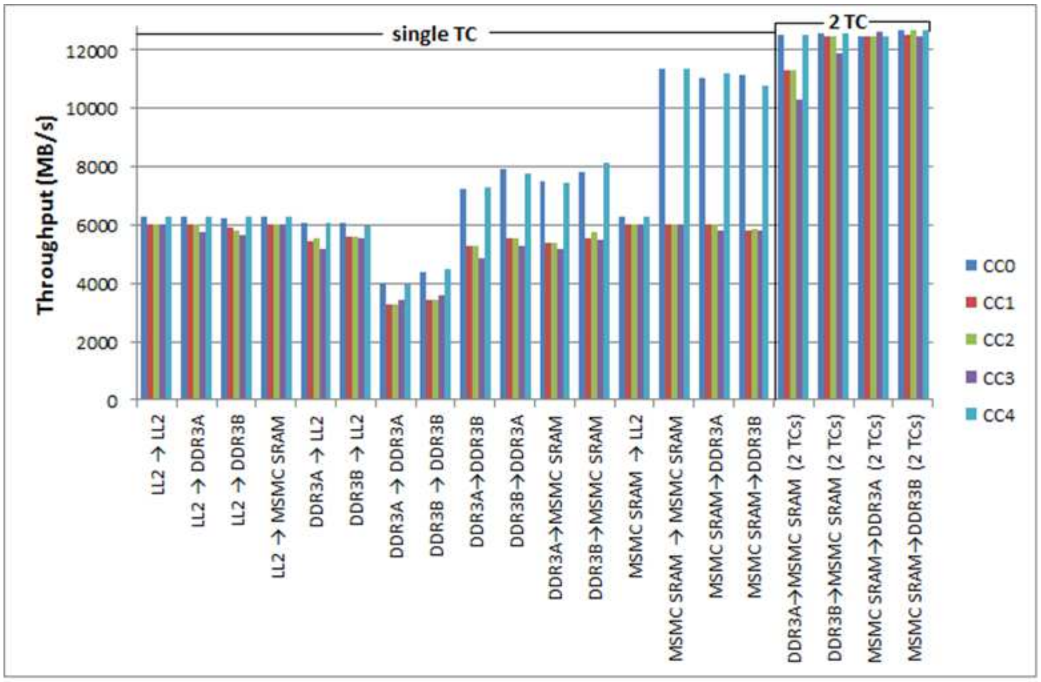


Figure 1. Throughput Comparison Between KeyStone-II TCs

For data transfer between MSMC SRAM and DDR3A/B, TCs on EDMACC0 and EDMACC4 should be used since they achieve much better throughput than other TCs. Due to the high latency of reading from DDR3 memory, a single EDMA TC is unable to issue enough read commands to keep the DDR3 EMIF command pipeline fully utilized which reduces the effective throughput when reading from DDR3A or DDR3B. EDMA is able to keep the command pipeline better utilized during writes thus resulting in higher performance compared to reads. As shown in Figure 1, two EDMA TCs are required to get closer to full DDR3 bandwidth (12800 MB/s for 64-bit, DDR3-1600) on reads or writes. Note that all DDR3 results were measured with a single rank DRAM configuration. In the case of dual rank, the DDR3 controller introduces 6 DDR3 clock cycles (corresponding to T_CSTA = 5 in the SDRAM Timing 4 Register) between accesses to different ranks. This reduces DDR3 bus efficiency by approximately 20% compared to single rank.

CC0 and CC4 TCs have a wider bus width (256-bit) compared to other TCs (128-bit) and are inherently higher performing than CC1, CC2, and CC3. CC1, CC2 and CC3 TCs have comparable performance to one another.

5.2 EDMA3 Complex Throughput

Various complex throughput tests were performed for measuring the EDMA throughput under different conditions. The following tests calculate the EDMA throughput without any background traffic in the CorePac.

The test process includes the following steps:

1. Get the transfer time for a payload of 32KB/channel (includes overhead).
2. Get the transfer time for 0 bytes - this is called a dummy transfer and closely approximates the overhead.
3. Subtract 2 from 1 to get the transfer time with overhead removed - t3.
4. Throughput = [(32KB * number of channels)/t3] * 1.2 GHz

The basis of these tests is to set up and trigger bidirectional (read and write) data transfers in parallel for different combinations of TCs of the EDMA. The channel controller is programmed such that each TC performs one read and one write transfer between the TC endpoints bidirectionally on two different logical channels. Once TCs are triggered, wait (poll) for the completion of all transfers and capture the throughput values.

The EDMA configuration uses A-synchronized transfers across all tests and TC combinations with ACNT = 32KB and BCNT = CCNT = 1 per logical channel.

Note that each of the TCs have bidirectional transfers —from endpoint A to endpoint B and the other from endpoint B to endpoint A on two different logical channels. For example, for the TC0 and TC1 testcase, there are a total of four transfers on four different logical channels.

The TeraNet allows for concurrent transfers between non-conflicting master/slave pairs and can support a very high total data rate across any endpoint. If transfers line up such that the source or destination memory is the same, then collisions occur and certain transactions will be blocked.

The throughput is limited by the lowest of the bus bandwidth, source throughput, and destination throughput. Each scenario is accompanied by a figure that highlights the primary bandwidth limiting factor for that scenario. Bidirectional arrows in the figures represent the bidirectional transfers mentioned above.

Note that the EDMA TC combinations used in the results are chosen such that they use independent paths between the TeraNets in order to create concurrent, non-conflicting transfers between source and destination slave memories. If TCs that map to the same inter-TeraNet bridge are used, the bridge will likely be the bandwidth limiting factor. The TC-to-bridge mapping between TeraNets 3A and 3C is shown in Table 5.

Note that TC0 through TC13 in this document correlate to the Data Manual nomenclature in the following way:

Table 5. TC-to-Bridge Mapping Between TeraNets 3A and 3C [1]

EDMA Masters	Br_2 (to TeraNet 3A)	Br_3 (to TeraNet 3A)	Br_5 (to TeraNet 3C)	Br_6 (to TeraNet 3C)	Br_7 (to TeraNet 3C)	Br_8 (to TeraNet 3C)	Br_9 (to TeraNet 3C)	Br_10 (to TeraNet 3C)	Br_SMS0 (to MSMC SRAM)	Br_SMS1 (to MSMC SRAM)	Br_SMS2 (to MSMC SRAM)	Br_SES0 (to DDR3)	Br_SES1 (to DDR3)	Br_SES2 (to DDR3)
EDMA TCs on TeraNet 3C														
EDMA_0_TC0_R	X	-	-	-	-	-	-	-	X	-	-	X	-	-
EDMA_0_TC0_W	X	-	-	-	-	-	-	-	X	-	-	X	-	-
EDMA_0_TC1_R	-	X	-	-	-	-	-	-	-	X	-	-	X	-
EDMA_0_TC1_W	-	X	-	-	-	-	-	-	-	X	-	-	X	-
EDMA_4_TC0_R	X	-	-	-	-	-	-	-	-	X	-	-	X	-
EDMA_4_TC0_W	X	-	-	-	-	-	-	-	-	X	-	-	X	-
EDMA_4_TC1_R	-	X	-	-	-	-	-	-	-	X	-	-	X	-
EDMA_4_TC1_W	-	X	-	-	-	-	-	-	-	X	-	-	X	-
EDMA TCs on TeraNet 3A														
EDMA_1_TC0_R	-	-	X	-	-	-	-	-	X	-	-	X	-	-
EDMA_1_TC0_W	-	-	X	-	-	-	-	-	X	-	-	X	-	-
EDMA_1_TC1_R	-	-	-	X	-	-	-	-	X	-	-	X	-	-
EDMA_1_TC1_W	-	-	-	X	-	-	-	-	X	-	-	X	-	-
EDMA_1_TC2_R	-	-	-	-	X	-	-	-	-	X	-	-	X	-
EDMA_1_TC2_W	-	-	-	-	X	-	-	-	-	X	-	-	X	-
EDMA_1_TC3_R	-	-	-	-	-	X	-	-	-	X	-	-	X	-
EDMA_1_TC3_W	-	-	-	-	-	X	-	-	-	X	-	-	X	-
EDMA_2_TC0_R	-	-	-	-	-	-	X	-	-	-	X	-	-	X
EDMA_2_TC0_W	-	-	-	-	-	-	X	-	-	-	X	-	-	X
EDMA_2_TC1_R	-	-	-	-	-	-	-	X	-	-	X	-	-	X
EDMA_2_TC1_W	-	-	-	-	-	-	-	X	-	-	X	-	-	X
EDMA_2_TC2_R	-	-	X	-	-	-	-	-	X	-	-	X	-	-
EDMA_2_TC2_W	-	-	X	-	-	-	-	-	X	-	-	X	-	-
EDMA_2_TC3_R	-	-	-	X	-	-	-	-	X	-	-	X	-	-
EDMA_2_TC3_W	-	-	-	X	-	-	-	-	X	-	-	X	-	-
EDMA_3_TC0_R	-	-	-	-	X	-	-	-	-	X	-	-	X	-
EDMA_3_TC0_W	-	-	-	-	X	-	-	-	-	X	-	-	X	-
EDMA_3_TC1_R	-	-	-	-	-	X	-	-	-	X	-	-	X	-
EDMA_3_TC1_W	-	-	-	-	-	X	-	-	-	X	-	-	X	-

[1] Legend: X = There is no connection between this TC and the bridge.
– = There is no connection between this TC and the bridge.

5.2.1 Scenario 1: EDMA Transfer Between CorePac0 L2 and CorePac1 L2

In this scenario, the endpoints used are CorePac0 L2 SRAM and CorePac1 L2 SRAM memory. There is contention among TCs as the same endpoint is used by both TCs for each of the bidirectional transfers. The TCs have to use the same SDMA/IDMA port of the same CorePac and the second transaction will have to wait until the first is complete.

So only one TC transfer is done at a time and the theoretical throughput is equivalent to (128 bits)/(8 bit/byte)×(1200M/3)=6400 MB/s.

The test case descriptions are as follows:

- Every TC sets up bidirectional data transfer between the same CorePac0 L2 and CorePac1 L2.
- **TC0 and TC1 test case:** Trigger TC0 and TC1 in parallel. After the completion of the TC0 and TC1 transfers in both directions, the steady state throughput is calculated.
- **TC2 and TC6 test case:** Trigger TC2 first and after a delta lag TC6 is triggered. After the completion of TC2 and TC6 transfers in both directions, the steady state throughput is calculated.

Table 6 shows the measured aggregate throughput of the TCs for this scenario.

Table 6. EDMA Throughput Between CorePac0 L2 and CorePac1 L2 Endpoints

Transaction on TCs	TC0 and TC1	TC2 and TC6
Wait for completion of	**Both**	**Both**
Theoretical Max throughput between TC endpoints (MBPS)	6400	6400
Aggregate throughput of TCs fired (MBPS)	6201	5935
Total transfer efficiency	97%	93%

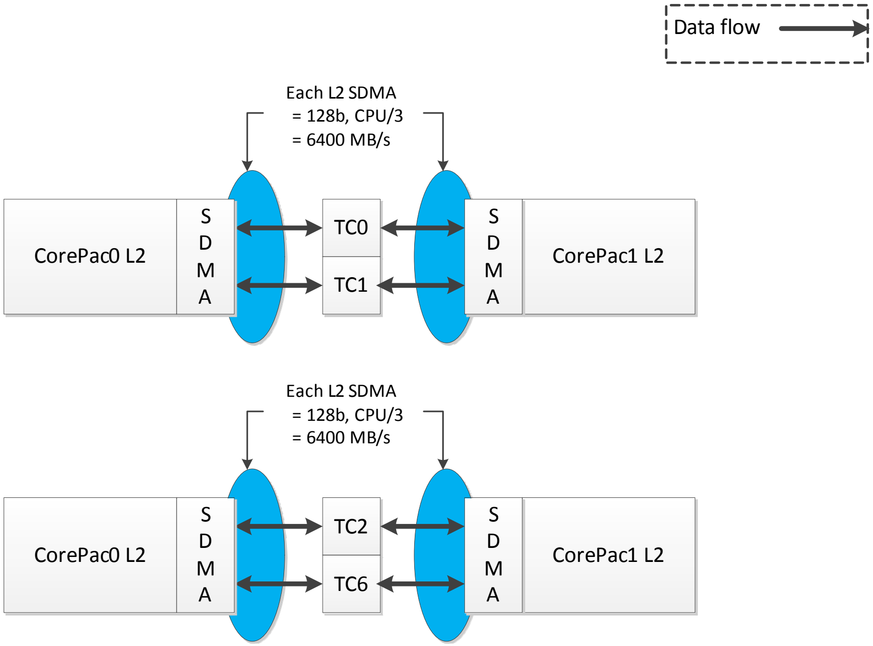


Figure 2. EDMA Transfer Between CorePac0 L2 and CorePac1 L2

5.2.2 Scenario 2: EDMA Transfer Between Different CorePac L2 and MSMC SRAM

In this scenario, each TC uses a different CorePac source address and there is no contention between TCs at the inter-TeraNet bridges. There may be some contention at the MSMC but it’s bandwidth exceeds the sum total of TC bandwidths and thus contention there may be minimal For example TC0 reads from CorePac0 L2, whereas TC1 reads from CorePac1 L2. So the total throughput for any two TCs would be 2× [(128 bits)/(8 bit/byte)×(1200M/3)] = 12800 MB/s and for any four TCs this would be 4× [(128 bits)/(8 bit/byte)×(1200M/3)] = 25600 MB/s.

The testcase setup details are as follows:

- **TC0 and TC1:** TC0 has bidirectional data transferred between CorePac0 L2 and MSMC. TC1 has bidirectional data transferred between CorePac1 L2 and MSMC.
- **TC2 and TC6:** TC2 has bidirectional data transferred between CorePac0 L2 and MSMC. TC6 has bidirectional data transferred between CorePac1 L2 and MSMC.
- **TC2, TC6, TC10, and TC12:** Each TC uses a different CorePac to transfer bidirectional data between MSMC and CorePac For example, TC2 uses CorePac0, TC6 uses CorePac1, TC10 uses CorePac2, and TC12 uses CorePac3.

Table 7 shows the measured aggregate throughput of the TCs for this scenario.

Table 7. EDMA Transfer Between Different CorePac L2 and MSMC SRAM

Transaction on TCs	TC0 and TC1	TC2 and TC6	TC2, TC6, TC10, and TC12
Wait for completion of	**Both**	**Both**	**All**
Theoretical Max throughput between TC endpoints (MBPS)	12800	12800	25600
Aggregate throughput of TCs fired (MBPS)	11843	11843	24272
Total transfer efficiency	93%	93%	95%

Figure 3 shows the measured aggregate throughput of the TCs for this scenario.

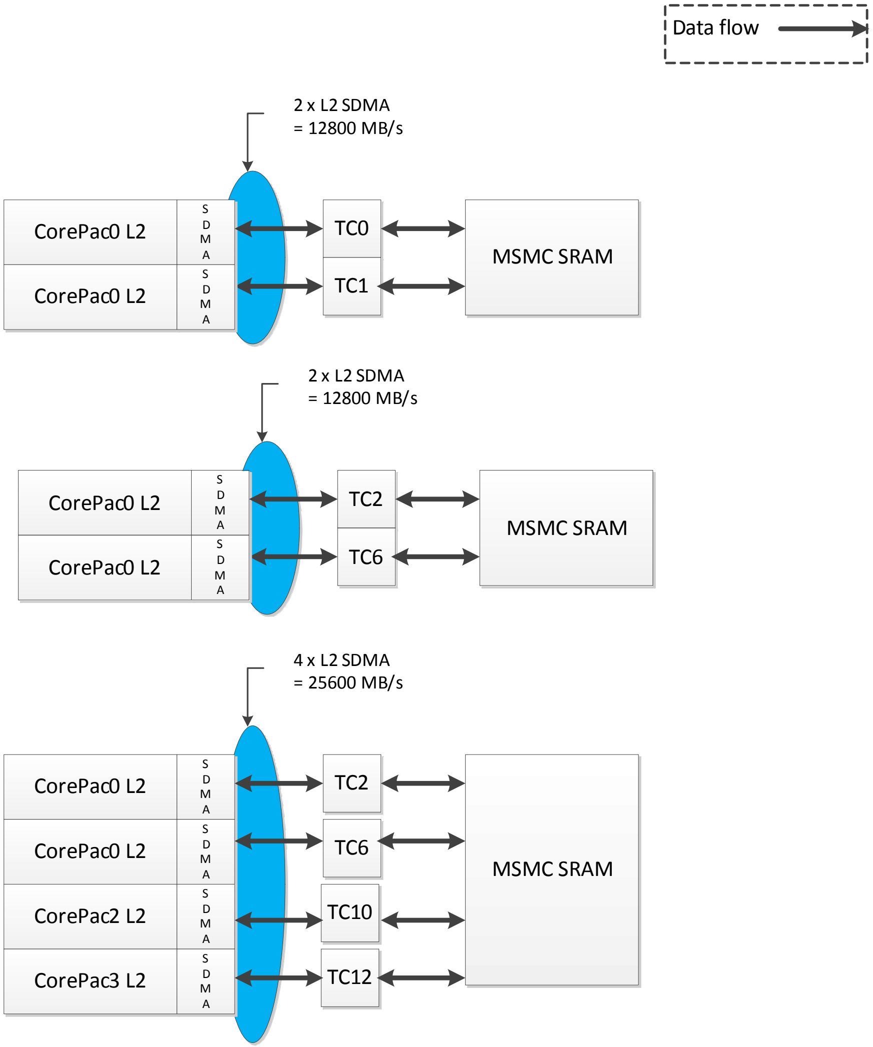


Figure 3. EDMA Transfer Between CorePac0 L2 and MSMC SRAM

5.2.3 Scenario 3: EDMA Transfer From MSMC SRAM to MSMC SRAM

In this scenario there is contention between TCs because the source/destination are the same for different TCs, so there is only one transfer happening at a time. Even though the theoretical throughput for MSMC is (256 bits)/(8 bit/byte)×(1200M)=38400MB/s, it cannot do both reads and writes at the same time. It has to wait for the previous read from MSMC to complete and then do the write operation to the MSMC. Therefore, the throughput will be effectively halved and is 38400/2 = 19200MB/s.

Note that for TC2 and TC6 use case, both TCs use two independent paths on TeraNet 3A which is a CPU/3 128 bit interface. The theoretical throughput is (128bits)/(8bit/byte)×2×(1200M)/3=12800 MB/s.

shows the measured aggregate throughput of the TCs for this scenario.

Table 8 shows the measured aggregate throughput of the TCs for this scenario.

Table 8. EDMA Throughput Between MSMC SRAM and MSMC SRAM

Transaction on TCs	TC0 and TC1	TC2 and TC6	TC2, TC4, and TC6	TC2, TC6, and TC10
Wait for completion of	**Both**	**Both**	**All**	**All**
Theoretical Max throughput between TC endpoints (MBPS)	19200	12800	19200	19200
Aggregate throughput of TCs fired (MBPS)	17329	11839	17331	16186
Total Transfer Efficiency	90%	92%	90%	84%

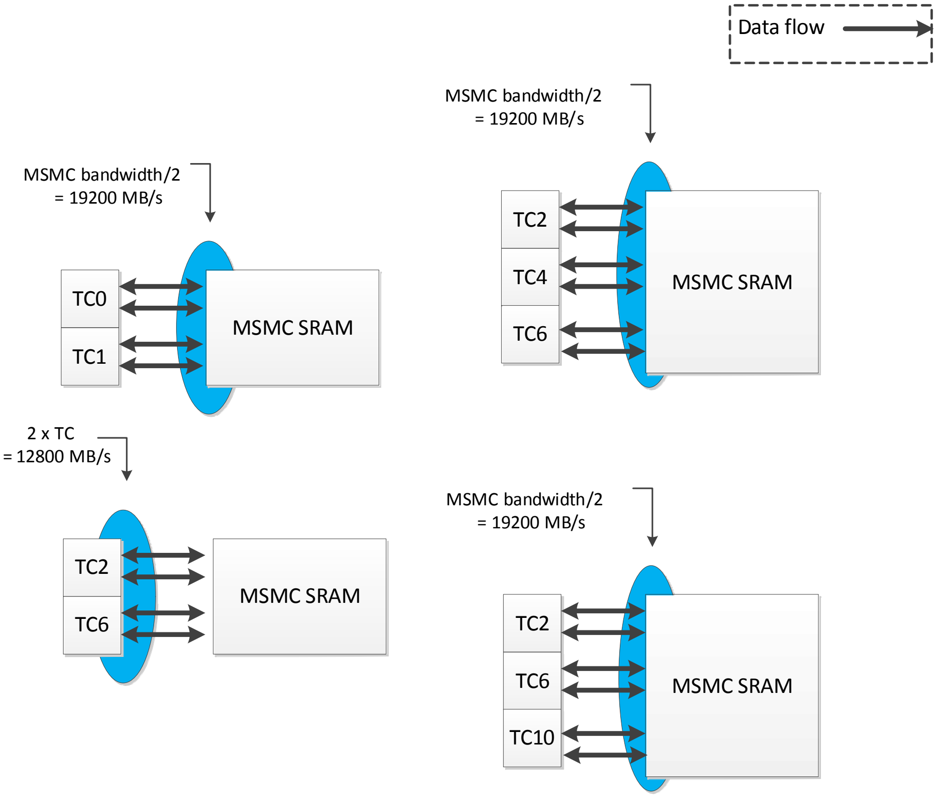


Figure 4. EDMA Throughput Between MSMC SRAM and MSMC SRAM

5.2.4 Scenario 4: EDMA Transfer From MSMC SRAM to DDR3

There is no contention between TCs at the inter-TeraNet bridges, but there may be some contention at the DDR3 memory with multiple TCs accessing the same memory endpoint. The throughput is limited by the maximum DDR3 bandwidth, which is (64 bits)/(8 bit/byte)×(800M)×2=12800MB/s.

The testcase setup details are as follows:

- **TC0 and TC1:** TC0 and TC1 have bidirectional data transferred between MSMC and DDR3A/DDR3B.
- **TC2 and TC6:** TC2 and TC6 have bidirectional data transferred between MSMC and DDR3A/DDR3B.
- **TC7, TC8 and TC9:** TC7, TC8 and TC9 have bidirectional data transferred MSMC RAM and DDR3A/3B.

Table 9 and Table 10 show the measured aggregate throughput of the TCs with DDR3A and DDR3B, respectively, for this scenario. Ignore Table 10 if your device does not support DDR3B.

Table 9. EDMA Throughput Between MSMC SRAM and DDR3A

Transaction on TCs	TC0 and TC1	TC2 andTC6	TC7, TC8 and TC9
Wait for completion of	**Both**	**Both**	**ALL**
Theoretical Max throughput between TC endpoints (MBPS)	12800	12800	12800
Aggregate throughput of TCs fired (MBPS)	10638	9925	12137
Total Transfer Efficiency	83%	78%	95%

Table 10. EDMA Throughput Between MSMC SRAM and DDR3B

Transaction on TCs	TC0 and TC1	TC2 andTC6	TC7, TC8 and TC9
Wait for completion of	**Both**	**Both**	**ALL**
Theoretical Max throughput between TC endpoints (MBPS)	12800	12800	12800
Aggregate throughput of TCs fired (MBPS)	10730	10291	12192
Total Transfer Efficiency	84%	80%	95%

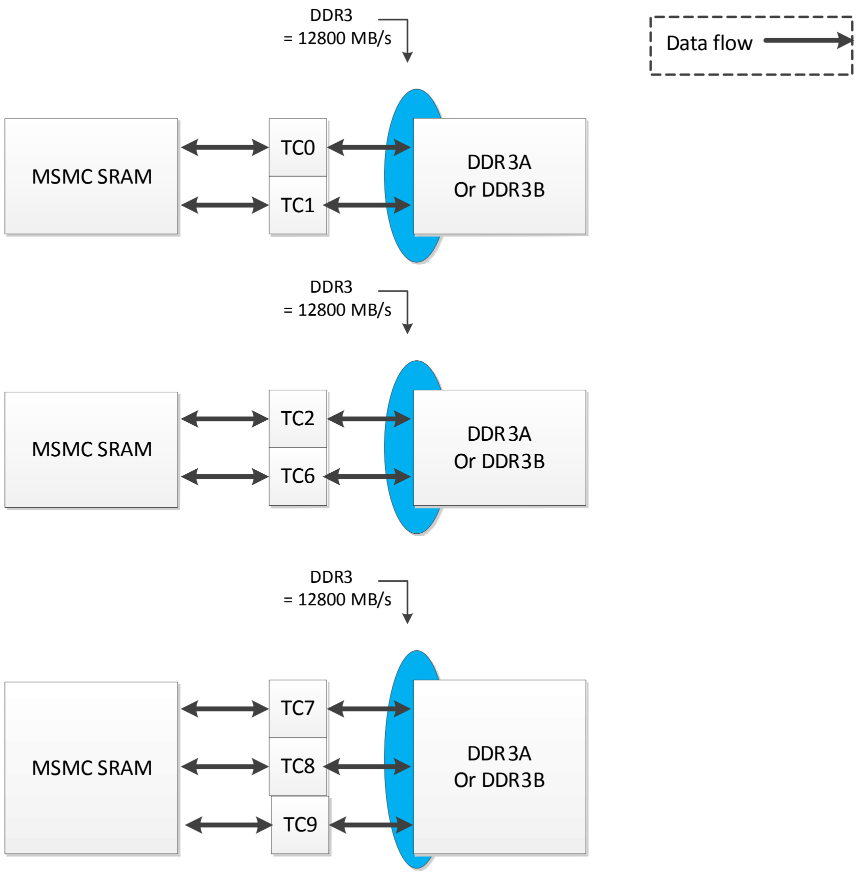


Figure 5. EDMA Transfer Between MSMC SRAM and DDR3A or DDR3B

5.2.5 Scenario 5: EDMA Transfer From DDR3 to DDR3

For bidirectional transfers between the same DDR3 endpoint: There is contention between TCs, but because the DDR3 controller cannot do a read from DDR3 and a write to DDR3 in parallel, the EDMA needs to wait until the previous operation is completed. The theoretical max DDR3 throughput of 12800MB/s will be halved and will be equal to 6400 MB/.

For bidirectional transfers between different DDR3 endpoints (DDR3A and DDR3B): Since the TCs can issue a read and write to different DDR3 endpoints in parallel, the theoretical max is not halved and still equals 12800 MB/s.

In both cases, the measured throughput is even lower due to the effects of read-write turnaround and page switch overheads.

The testcase setup details are as follows:

- Every TC sets up bidirectional data transfer between DDR3A and DDR3A SDRAM.
- Every TC sets up bidirectional data transfer between DDR3B and DDR3B SRAM.
- Every TC sets up bidirectional data transfer between DDR3A and DDR3B SDRAM.

Table 11 and Table 12 show the measured aggregate throughput of the TCs for this scenario. Ignore Table 12 and Table 13 if your device does not support DDR3B.

Table 11. EDMA Throughput Between DDR3A and DDR3A

Transaction on TCs	TC0 and TC1	TC2 and TC6	TC10 and TC11
Wait for completion of...	Both	Both	Both
Theoretical Max throughput between TC endpoints (MBPS)	6400	6400	6400
Aggregate throughput of TCs fired (MBPS)	4341	4108	3954
Total Transfer Efficiency	84%	80%	95%

Table 12. EDMA Throughput Between DDR3B and DDR3B

Transaction on TCs	TC0 and TC1	TC2 and TC6	TC10 and TC11
Wait for completion of...	Both	Both	Both
Theoretical Max throughput between TC endpoints (MBPS)	6400	6400	6400
Aggregate throughput of TCs fired (MBPS)	4078	3629	3330
Total Transfer Efficiency	64%	57%	52%

Table 13. EDMA Throughput Between DDR3A and DDR3B

Transaction on TCs	TC0 and TC1	TC2 and TC6	TC10 and TC11
Wait for completion of...	Both	Both	Both
Theoretical Max throughput between TC endpoints (MBPS)	12800	12800	12800
Aggregate throughput of TCs fired (MBPS)	10334	9806	9240
Total Transfer Efficiency	81%	77%	72%

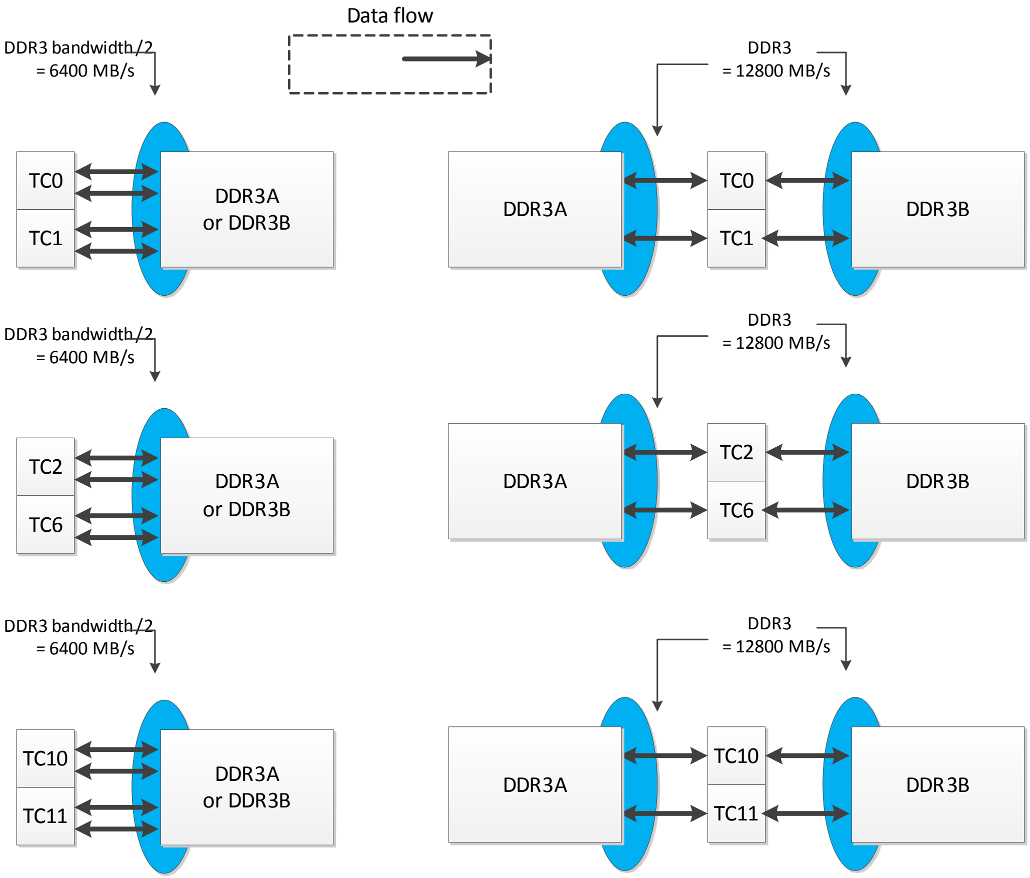


Figure 6. EDMA Throughput Between DDR3A and DDR3A

5.2.6 Scenario 6: EDMA Transfer From Different CorePac L2 to DDR3

In this scenario, each TC uses a different CorePac source address and there is no contention between TCs at the inter-TeraNet bridges. However, there is contention among multiple TCs at the DDR3 memory. The throughput is bounded by the DDR3A/DDR3B maximum throughput, which is 12800MB/s.

The testcase setup details are as follows:

- **TC0 and TC1:** TC0 has bidirectional data transferred between CorePac0 L2 and DDR3A/3B. TC1 has bidirectional data transferred between CorePac1 L2 and DDR3A/3B.
- **TC2 and TC6:** TC2 has bidirectional data transferred between CorePac0 L2 and DDR3A/3B. TC6 has bidirectional data transferred between CorePac1 L2 and DDR3A/3B.
- **TC10 and TC11:** TC10 has bidirectional data transferred between CorePac0 L2 and DDR3A/3B. TC11 has bidirectional data transferred between CorePac1 L2 and DDR3A/3B.

Table 14 and Table 15 show the measured aggregate throughput of the TCs for this scenario. Ignore Table 15 if your device does not support DDR3B.

Table 14. EDMA Throughput Between Different CorePac L2s to DDR3A

Transaction on TCs	TC0 and TC1	TC2 and TC6	TC10 and TC11
Wait for completion of...	**Both**	**Both**	**Both**
Theoretical Max throughput between TC endpoints (MBPS)	12800	12800	12800
Aggregate throughput of TCs fired (MBPS)	10528	10135	9849
Total Transfer Efficiency	82%	79%	77%

Table 15. EDMA Throughput Between Different CorePac L2s to DDR3B

Transaction on TCs	TC0 and TC1	TC2 and TC6	TC10 and TC11
Wait for completion of...	Both	Both	Both
Theoretical Max throughput between TC endpoints (MBPS)	12800	12800	12800
Aggregate throughput of TCs fired (MBPS)	10518	10520	10311
Total Transfer Efficiency	82%	82%	81%

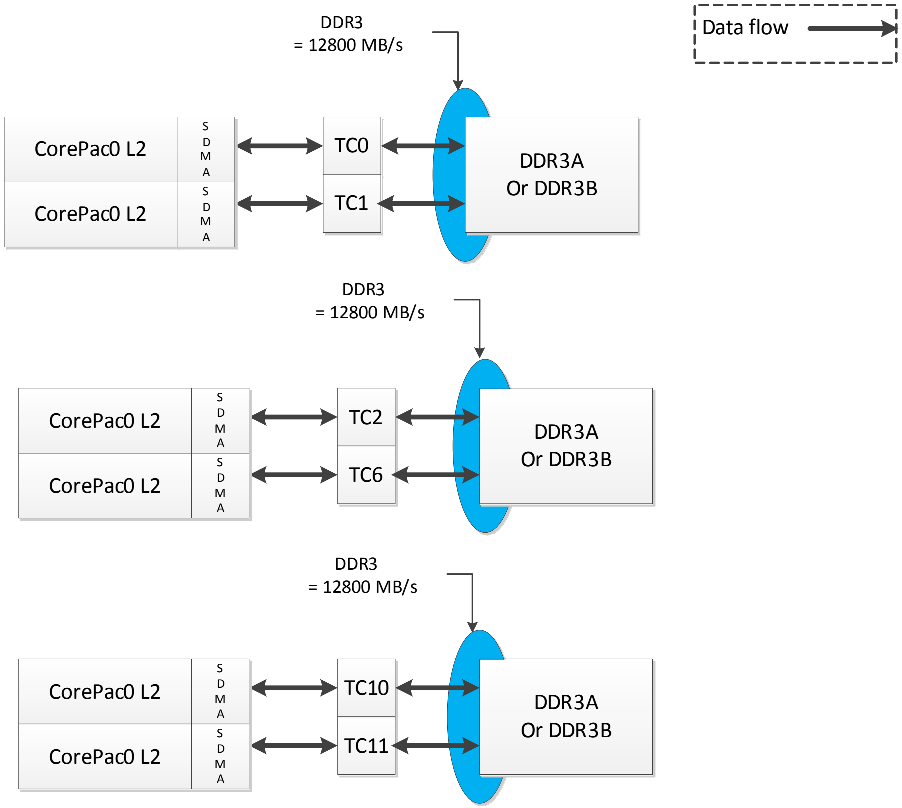


Figure 7. EDMA Throughput Between Different CorePac L2s to DDR3A

5.3 *Impact of Various Parameters on EDMA3 Throughput*

This section discusses the impact of various parameters on EDMA3 throughput. EDMA3 is used for data movement in various applications. Some applications might implement single-dimensional (contiguous) bulk transfers. Others might execute multi-dimensional, non-linear transfers such as matrix transposition. The transfer geometry can greatly impact the efficiency of EDMA transfers. So can the choice of EDMA CC/TCs and source/destination memory endpoints. Many of these transfers perform memory paging between internal memory and external memory like DDR3 in order to make it available to the CPU before a real time processing deadline expires. Thus, it is important to understand how the various parameters impact EDMA transfer efficiency.

5.3.1 Impact of Total Transfer Size

To make full utilization of the available EDMA bandwidth, it is important to fully utilize the bus width and burst sizes. In order to optimize the EDMA bandwidth, the ACNT x BCNT bytes (for linear) and ACNT bytes (for non-linear):

- Must be larger than the TC bus width. The bus width is 32 bytes for EDMACC0 and EDMACC4 TCs and 16 bytes for other TCs.
- Must be larger than the default burst size (DBS) of the EDMA TC. DBS is 64 bytes for EDMACC3 TCs and 128 bytes for all others.

- Must be larger than the TC FIFO size. This is 1024 bytes for all TCs.

Figure 8 shows the impact on throughout for a linear transfer, as it crosses the various stages outlined above. EDMACC0 TC0 performing a data transfer from MSMC SRAM to CorePac0 L2 is chosen to demonstrate. A log scale is chosen for the X-axis to effectively visualize throughput over a large range of transfer sizes.

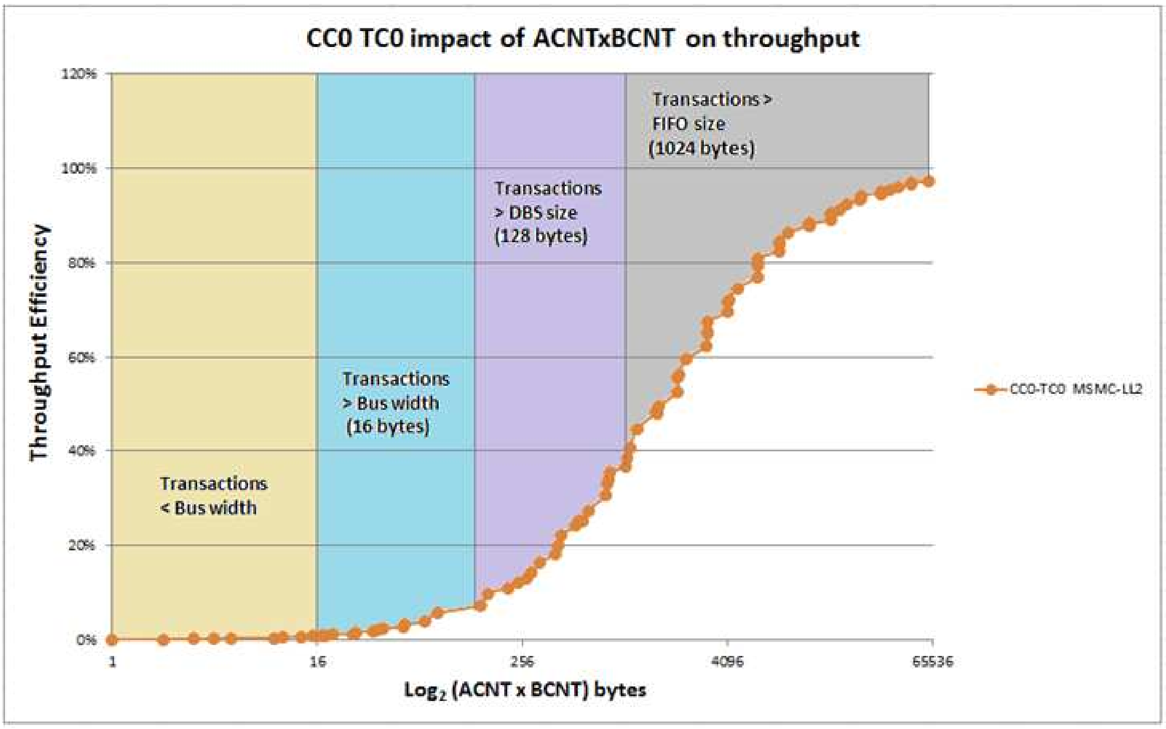


Figure 8. Impact of Transfer Size on Throughput

5.3.2 Impact of Index

Index significantly impacts EDMA throughput. In Figure 9, EDMACC0 TC0 performing data transfers from MSMC SRAM to CorePac0 L2 is used to demonstrate the impact.

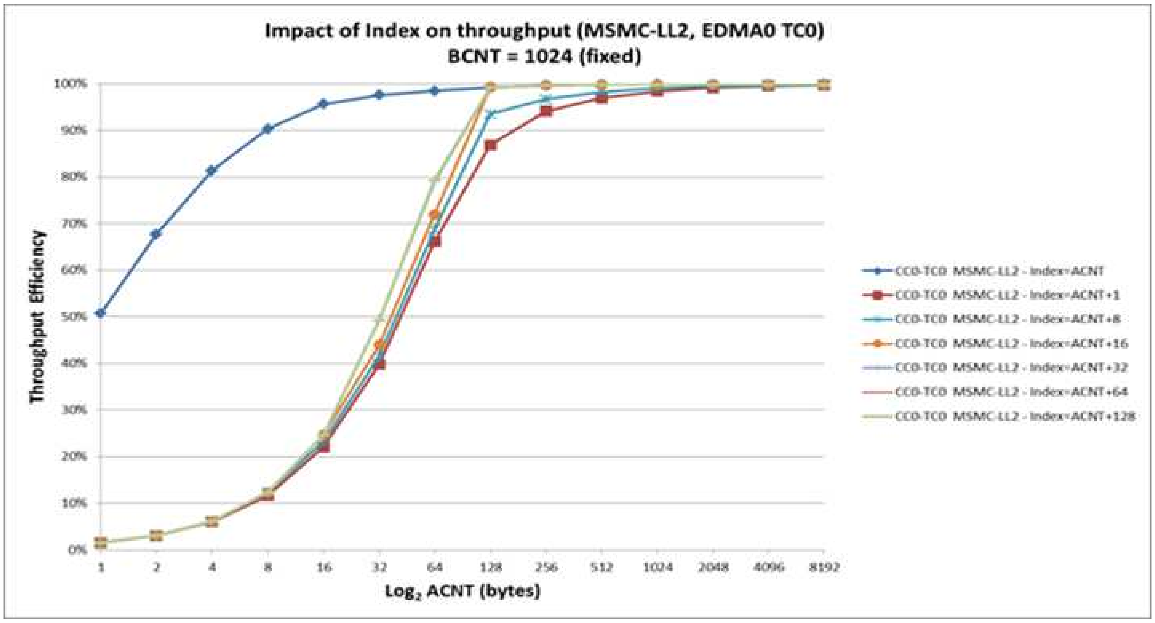


Figure 9. Impact of Index on Throughput

A linear transfer (Index=ACNT) performs best because EDMA optimizes the transfer by fully utilizing TC bandwidth. The starting address for MSMC SRAM programmed in the EDMA PaRAM is 64-byte aligned. Incrementing the index in (ACNT + power-of-2) also increases the percentage of SRAM sub-banks that are fully utilized by the EDMA burst, thus increasing the bandwidth efficiency (a sub-bank in MSMC SRAM lies at every 32-byte boundary and is 32-bytes wide). The sub-bank utilization is the worst for an odd index, making it the worst performing condition. For ACNT = 16 bytes or less, the sub-bank utilization is approximately the same for all conditions, making the plots converge. The pipelined bus architecture utilizes the bus infrastructure more efficiently as ACNT increases. This minimizes the significance of sub-bank inefficiency and increases the overall throughput efficiency until all plots eventually converge. For details on SRAM bank architecture, see the *Multicore Shared Memory Controller (MSMC) User's Guide* (SPRUHJ6).

5.3.3 Impact of Address Alignment

Address alignment may slightly impact the performance. The default burst size of EDMA3 is 64bytes or 128 bytes. The EDMA transfer controller always generates read or write commands that are less than or equal to DBS size. If the source or destination addresses are not aligned on a DBS-sized boundary, EDMA will generate an extra burst to handle the unaligned head and tail data. This overhead may be ignored for large transfers (typically larger than TC FIFO size).

5.4 General Recommendations

Observations from previous sections lead to some general guidelines. In order to maximize EDMA bandwidth efficiency:

- DO:
  - Use linear transfers
  - Use power-of-2 sized block sizes
  - Use block sizes larger than TC FIFO size
  - Use block sizes larger than TC DBS size
  - Use block sizes larger than TC bus width size

  - Use block sizes larger than TC bus width size
  - Use EDMACC0 and EDMACC4 TCs for transfers involving MSMC SRAM, DDR3A or DDR3B.
- AVOID:
  - Using non-linear transfers
  - Using odd-sized source or destination indices
  - Using block sizes smaller than TC bus width/DBS/FIFO size

6 PCIe

This section discusses the throughput performance of the Keystone II PCIe peripheral. The Keystone II PCIe peripheral is connected to the TeraNet through a 128-bit bus. The peripheral operates at a frequency of DSP/3; however, the I/O clock frequency for this module is derived from the PCIe SerDes PLL. The PCIe peripheral supports the PCIe v2 standard, which has a maximum theoretical throughput of 5 gigabits per second per link. Keystone II devices support two PCIe links, each capable of transferring data five gigabits per second. Together, these two PCIe links have a maximum theoretical throughput of s per second.

It is important to note that the per second data rate includes the overhead due to the physical layer, the data link layer (DLL), and the transaction layer packet (TLP). Due to overhead, the raw data that is able to be transferred is less s per second. The remainder of this section discusses:

- Main Factors Affecting PCIe Throughput Performance
- How to Achieve Maximum Throughput with the PCIe Peripheral
- Measured Throughput Performance

6.1 Main Factors Affecting PCIe Throughput Performance

This section discusses the main factors that affect the performance of the PCIe peripheral. To achieve the maximum throughput possible using the Keystone II PCIe peripheral, the following factors must be kept in mind:

- Overhead Considerations
- Packet Size Considerations
- Packet Size Considerations

6.1.1 Overhead Considerations

The overhead discussed in this section is related to the PCIe protocol itself, and is not due to the Keystone II PCIe implementation. For each PCIe data transfer TLP packets have 20-28 bytes of overhead due to encapsulation. There is also overhead for flow control and acknowledge DLLP packets, which adds 8 bytes of overhead per packet. All data sent over the bus, is encoded using the 8b/10b encoding scheme, which adds an extra 20% overhead. The exact number of bytes of overhead that are required per packet is dependent on the size of the data payload, and the frequency of the DLLP packets.

6.1.2 Packet Size Considerations

For each packet transferred over the PCIe protocol, the overhead associated with transferring the data should be taken into account. For more information on PCIe overhead, see Section 6.1.1. Since the overhead is added on a per packet basis, the amount of overhead can be reduced by making the data payload as large as possible for all packets transferred.

For outbound transfers, the Keystone II PCIe peripheral is able to transfer packets of up to 128 bytes, excluding overhead. Therefore, transferring packets with 128 bytes of data payload will achieve the best throughput. For inbound data transfers, the PCIe peripheral is capable of handling payload sizes larger than 128 bytes, which can further reduce overhead and increase throughput performance. Impact of packet size on performance is discussed in the Measured Throughput section.

NOTE: For outbound transfers: The packet size is different from the data transfer size. For outbound transfer, the maximum payload size of a packet is 128 bytes. However, data transfers over 128 bytes are supported by the PCIe peripheral. Data transfers over 128 bytes are broken into multiple packets.

NOTE: For inbound transfers: PCIe can take advantage of 256 byte inbound size as long as the external device also has 256 bytes (or more) of outbound payload size. This is not possible if the external device is a KeyStone device, because the maximum outbound payload size is only 128 bytes. That limits the maximum inbound packet size to 128B.

6.1.3 EDMA Considerations

When using EDMA to transmit data to the Keystone II PCIe, or when using EDMA to receive data from the Keystone II PCIe, the transfer controller (TC) that is used should be taken into consideration. Not all transfer controllers use the same data burst size (DBS), and the size of the data burst can have an impact on the performance of the PCIe peripheral. Since the PCIe peripheral supports data payload sizes of up to 128 bytes, if available, a EDMA TC that also supports 128-byte data burst sizes should be chosen. By matching the PCIe data payload size to the EDMA TC data burst size, the PCIe peripheral can send or receive the maximum possible payload size, thereby reducing overhead as much as possible.

When using a 128-byte data payload, the overhead will be introduced every 128 bytes of payload. If using an EDMA TC with a data burst size below 128 bytes, then more overhead will be introduced. For example, if an EDMA TC with a 64-byte data burst size is chosen, then the PCIe will use 64-byte payloads, and packet overhead will be introduced for every 64 bytes of payload data. When transferring 128 bytes, using two 64-byte data payloads will introduce twice as much overhead as a one 128-byte data payload. For more information on PCIe overhead, see Section 6.1.1. Impact of EDMA TC selection on performance is discussed in Section 6.3.

6.2 How to Achieve Maximum Throughput With the PCIe Peripheral

To maximize the throughput of the Keystone II PCIe peripheral, the peripheral should be programmed while keeping in mind the ‘‘Main Factors Affecting PCIe Throughput Performance’’. To achieve maximum performance for the Keystone II PCIe peripheral, the two main considerations should be to reduce packet overhead (maximize data payload) and to reduce the amount of time related to EDMA transactions.

To reduce overhead, the data payload should be as large as possible. The larger the data payload size, the less packet overhead is introduced, and the more the performance is increased. For outbound transactions, the maximum data payload size is 128 bytes, so payloads as close to 128 bytes as possible are desired. Furthermore, to ensure that the EDMA transfer controller provides 128-byte data payloads to the PCIe peripheral, an EDMA transfer controller that supports 128-byte data burst size should be used.

For inbound transactions, the PCIe v2 specification permits data payloads up to 4096 bytes; however, in Keystone II devices, the maximum PCIe data payload size is limited to 256 bytes for inbound transactions (compared to 128 bytes for outbound). For details, see the *KeyStone Architecture Peripheral Component Interconnect Express (PCIe) User's Guide* (SPRUGS6). As explained in the Note above, inbound packet size is limited to 128 bytes if the external device is a KeyStone device. The closer to maximum payload size, the better the throughput performance. Another way to reduce overhead is to reduce packet headers as much as possible. One example may be to remove optional headers, such as the TLP ECRC, if this field is not required by the application. ECRC generation can be disabled by clearing ECRC_GEN_EN bit field in PCIE_ACCR register.

Lastly, overhead can be reduced by reducing DLLP packets. If not required by the application, DLLP flow control packets can be disabled by programming the FC_DISABLE field in the LANE_SKEW register. IF DLLP acknowledge packets are not required by the application, then the transmission of these packets can be disabled by programming the ACK_DISABLE field in the LANE_SKEW register. If acknowledge packets are required by the application, the frequency of the acknowledge packets can be configured to the lowest rate required by the application by programming the ACK_FREQ register.

6.3 Measured Throughput Performance

The measurements mentioned in this section were collected on a 66AK2E05 device, which has two PCIe controllers (PCIe0 and PCIe1). PCIe0 data applies to all Keystone-II devices. PCIe1 data applies to 66AK2Ex and AM5K2Ex devices only.

Performance data for the PCIe peripheral is captured in and . For all performance tests, both links of the Keystone II PCIe module were connected directly with another Keystone II PCIe module. One module was configured as the RC and the other device was configured as the EP. For the performance tests, all transactions were outbound transfers initiated by the RC. Some results include bidirectional transfers (on both Keystone-II devices). The tests were measured with MSMC SRAM and DDR3A memory endpoints, and with 2048, 4096, 8192, 16384 and 32768-byte data payload transfer sizes. The published throughput numbers are the result of trending the measured throughput over the various payload transfer sizes.

The performance tests were conducted using TCs from all EDMAs spanning 64-byte and 128-byte data burst sizes. All packets with data payload used PHY, DLLP, and TLP headers. Each TLP packet contained 20-byte overhead of header and footer. Thus, the theoretical data throughput for 2 lanes configured at 5Gbps after removing the overhead and 8b/10b encoding is (packet size in bytes)/(packet size in bytes + 20) x (8/10) x (2 x 5Gbps) = 6.10 Gbps for 64-byte packets and 6.92 Gbps for 128-byte packets. This is plotted on the graphs. An 8-byte DLLP packet was sent after each packet containing a data payload. Performance data with all overhead excluded is shown in Section 6.3.1 and Section 6.3.2.

6.3.1 PCIe Read Throughput

The throughput shown in this section is the data payload only, and does not include any overhead due to PHY, DLLP, or TLP headers, and does not include overhead due to 8b/10b encoding.

For this test, read transactions were executed between two Keystone II PCIe devices. Device1 was configured as the RC and Device2 was configured as the EP.

For unidirectional reads, the read transactions were all outbound reads, where Device1 read data from Device2. For these transactions, the read request was initiated by Device1. When Device2 received the read request, it used the PCIe master to transfer the read data from memory to the PCIe module. When Device1 received the read data from Device2, the PCIe slave in Device1 provided the data to EDMA to transfer the data from the PCIe module to memory. PCIe0 of Device1 is connected to PCIe0 on Device 2 (same for PCIe1).

For bidirectional transfers, PCIe on both devices initiated read requests simultaneously.

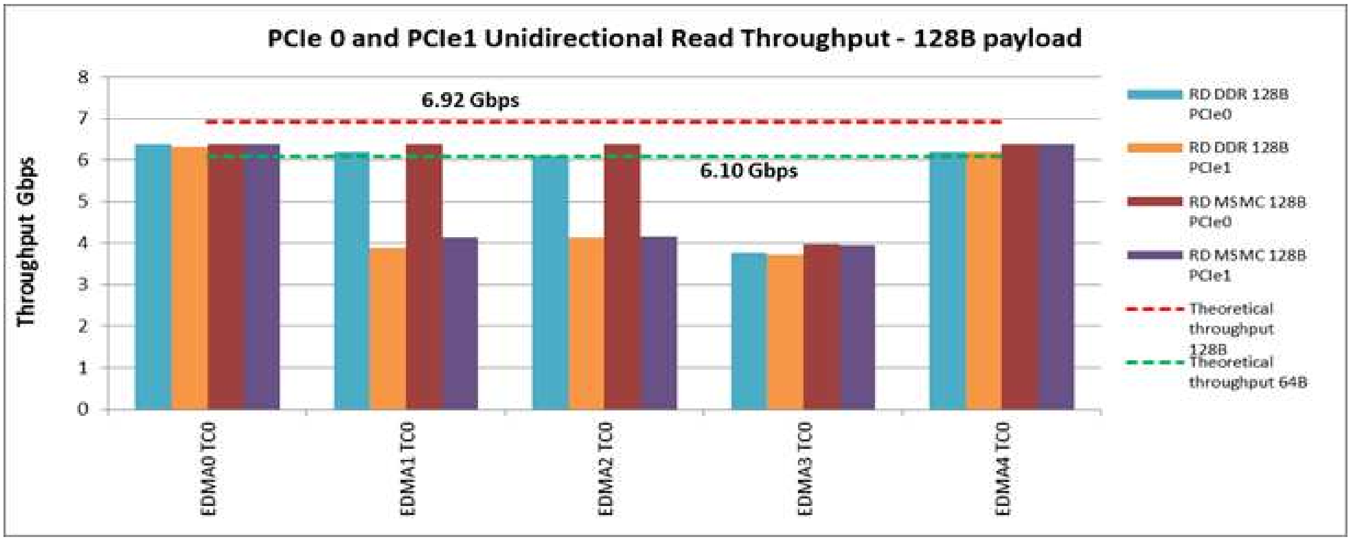


Figure 10. Impact of EDMA Selection on Throughput – Unidirectional Reads

Figure 10 shows the impact of EDMA TC selection on the read throughput when remote device returns data with 128 byte payload size. Overall, EDMA0 and EDMA4 perform best for both PCIe0 and PCIe1. EDMA1 and EDMA2 perform similar to EDMA0 & EDMA4 for PCIe0 but do not perform well with PCIe1. This is because EDMA1/EDMA2 and PCIe1 slave lie on different TeraNets which causes 128 byte packets to fragment into 64 byte packets while performing the inter-TeraNet transaction to PCIe1 slave. This increases overhead and reduces performance. PCIe0 is not impacted since PCIe0 slave lies on the same TeraNet as EDMA1 and EDMA2; fragmentation is thus avoided. EDMA3 performance is lower for both PCIe0 and PCIe1 because it has a smaller DBS size (64 bytes) compared to other EDMAs (128 bytes) resulting in extra overhead.

Some small variations in the results shown may be attributed to measurement accuracy.

Figure 11 shows the plots in Figure 10 combined with data when the remote device returns data with 64 byte packet size. A 64 byte payload size does not allow EDMAs with 128 byte DBS (EDMA0/1/2/4) to take advantage of the full burst size, resulting in lower throughput compared to 128 byte payload size. EDMA3 results are unaffected since it already has a 64-byte DBS.

Some small variations in the results shown may be attributed to measurement accuracy

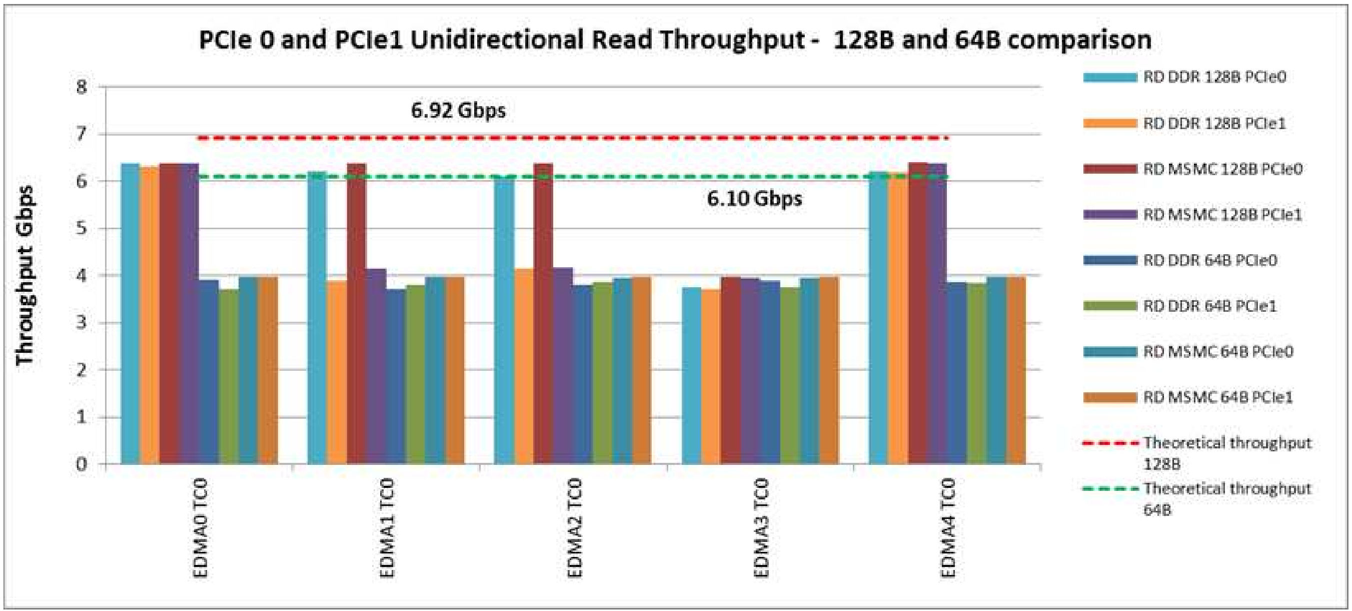


Figure 11. Comparing 128B and 64B Payload Size – Unidirectional Reads

Figure 12 and Figure 13 are identical to Figure 10 and Figure 11, respectively, expect that bidirectional transfers are executed over the PCIe interface. Some small variations in the results shown may be attributed to measurement accuracy. Bidirectional transfers slow down the PCIe throughput.

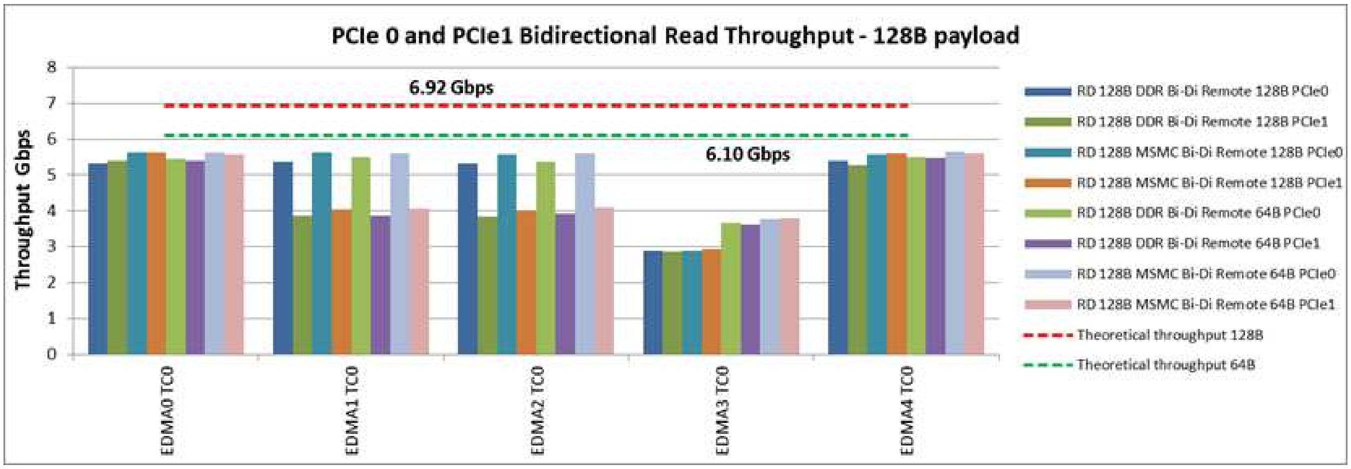


Figure 12. Impact of EDMA Selection on Throughput – Bidirectional Reads

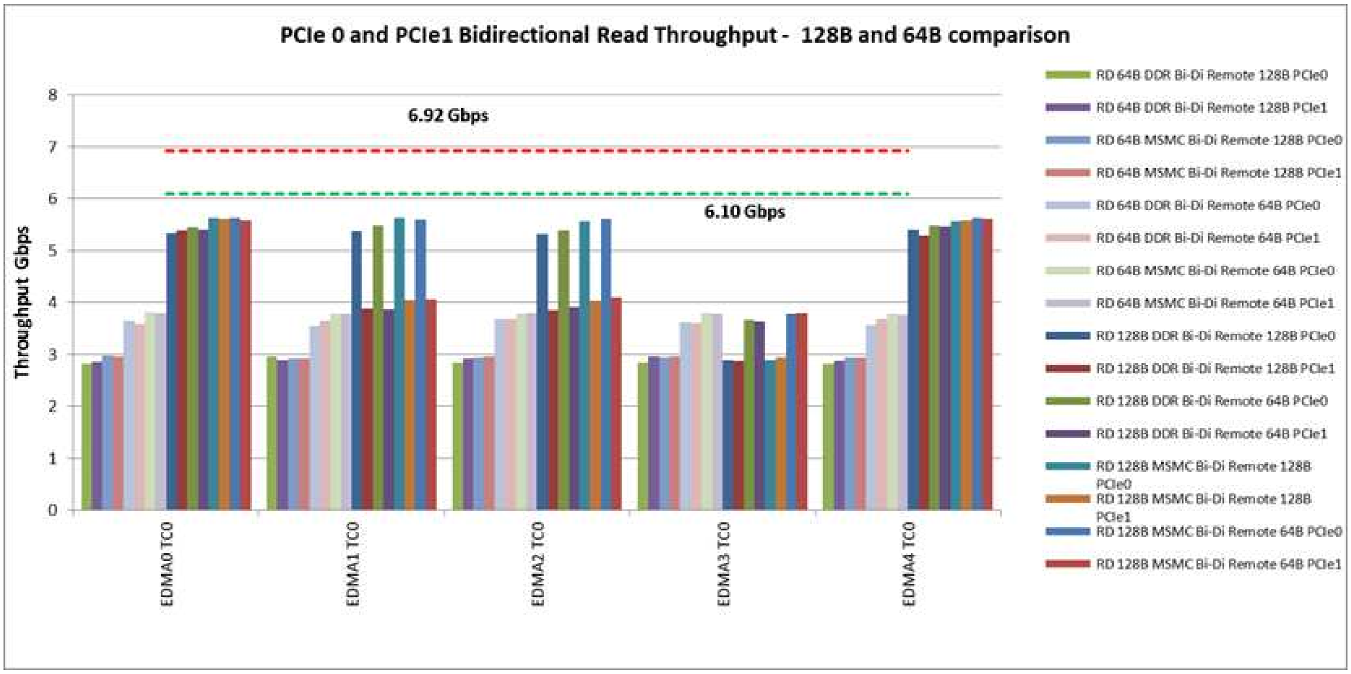


Figure 13. Comparing 128B and 64B Payload Size – Bidirectional Reads

The following broad recommendations can be made to optimize read throughput on PCIe:

- • Use 128 byte payload size
- • For both PCIe0 and PCIe1, use EDMA0 or EDMA4 where possible
- • Avoid 64 byte payload size
- • Avoid EDMA1 and EDMA2 for PCIe1
- • Avoid EDMA3 for both PCIe and PCIe1

6.3.2 PCIe Write Throughput

The throughput shown in this section is the data payload only, and does not include any overhead due to PHY, DLLP, or TLP headers, and does not include overhead due to 8b/10b encoding. For this test, write transactions were executed between two Keystone II PCIe devices. Device1 is configured as the RC and Device2 is configured as the EP.

For unidirectional writes, the write transactions were all outbound writes, where Device1 writes data to Device2. For these transactions, the write request was initiated by Device1. Device1 used EDMA to transfer data from memory to the PCIe slave, which was transferred over the PCIe interface to Device2. When Device2 received the write data, the PCIe master was used to transfer the data to memory. For bidirectional transfers, PCIe on both devices initiated write requests simultaneously.

“Diff ports” is a special case where PCIe0 on Device1 is connected to PCIe1 on Device2 and PCIe1 on Device1 is connected to PCIe0 on Device2. For all other cases, PCIe0 of Device1 is connected to PCIe0 on Device 2 (same for PCIe1).

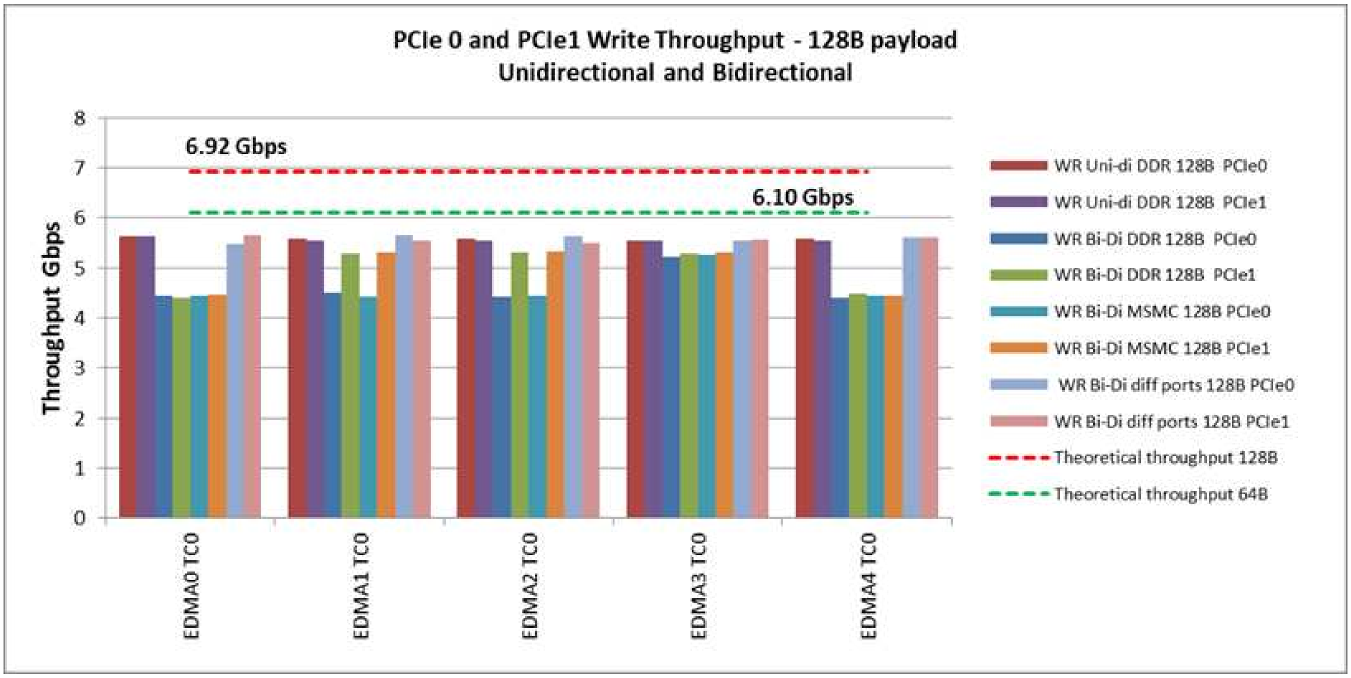


Figure 14. Impact of EDMA Selection on Write Throughput

Figure 14 shows the unidirectional and bidirectional write throughput for PCIe0 and PCIe1 for various EDMA selections. All EDMAs perform equally well for unidirectional write throughput on both PCIe0 and PCIe1. Bidirectional transfers slow down the PCIe0 throughput when EDMA0/1/2/4 are used. However, EDMA3 can be used without slowing down PCIe0 bidirectional write throughput. PCIe1 bidirectional throughput degrades when using EDMA0 or EDMA4, but performs well for EDMA1, EDMA2 and EDMA3. Bidirectional throughput when PCIe are connected to different ports (PCIe0 to PCIe1 and vice-versa) performs well regardless of EDMA selection.

Some small variations in the results shown may be attributed to measurement accuracy.

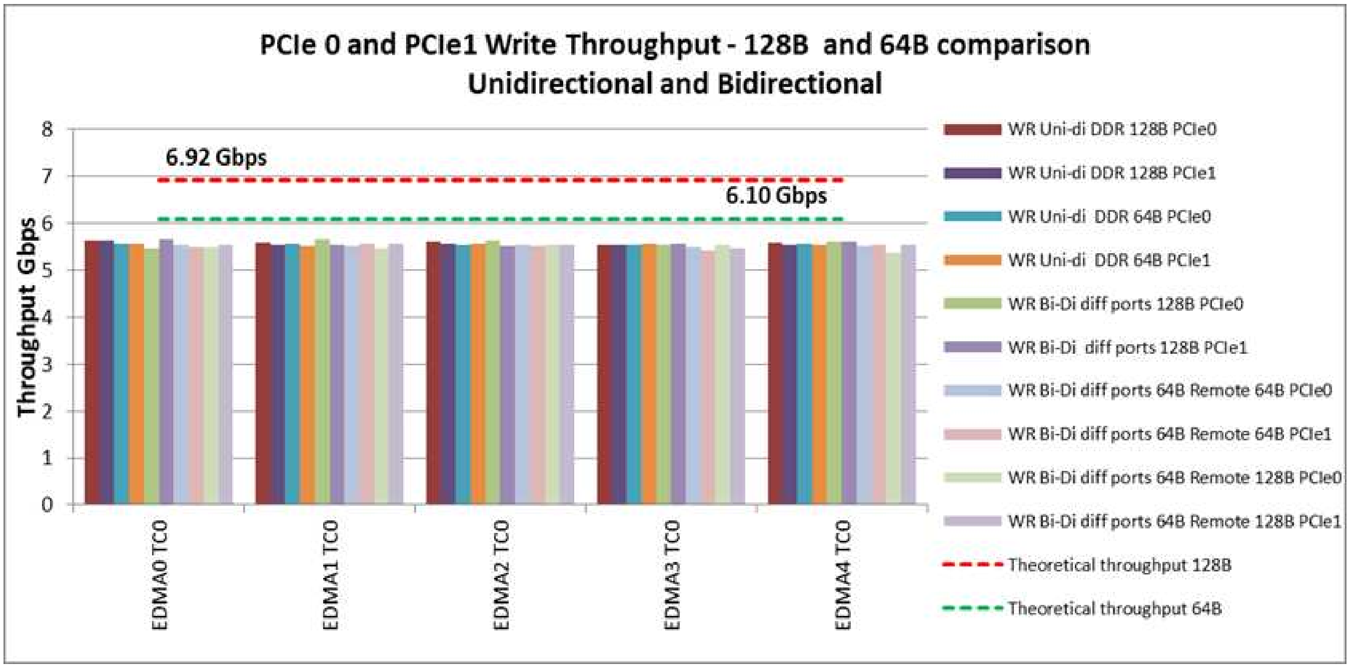


Figure 15. Comparing 128B and 64B Payload Size

As seen in Figure 15, the write throughput does not change with the payload size due to an internal feature of the PCIe subsystem. This feature also results in write throughput being lower than the read throughput for 128 byte payload size.

The following broad recommendations can be made to optimize write throughput on PCIe:

- For unidirectional or bidirectional throughput (both PCIe0 and PCIe1), use EDMA3 where possible. For PCIe1, EDMA1 and EDMA2 can also be used.
- For bidirectional throughput (both PCIe0 and PCIe1) avoid EDMA0 and EDMA4. For PCIe0, also avoid EDMA1 and EDMA2.

7 SRIO Throughput

Serial RapidIO (SRIO) is a high-bandwidth system level interconnect. It is a packet switched interconnect intended for high speed chip-to-chip and board-to-board communication. SRIO supports DirectIO transfers and Message Passing transfers.

The RapidIO specification defines four different bandwidths for each differential pair of I/O signals: 1.25, 2.5, 3.125, and 5 Gbps. Due to 8-bit/10-bit encoding overhead, the effective data bandwidths per differential pair is 1, 2, 2.5, and 4 Gbps, respectively. A 1x port is defined as one TX and one RX differential pair. A 4x port is a combination of four of these pairs.

7.1 DirectIO (LSU) Operation

The DirectIO (Load/Store) module serves as the source of all outgoing directIO packets. In this operation, the RapidIO packet contains the specific address where the data should be stored or read in the destination device. There are 8 LSUs in total.

The DirectIO tests were measured on a 66AK2H12 operating at 1 GHz. L2 memory endpoints were used for both input and output data. The throughput test process includes the following:

- Separate data transfers are initiated for NWRITE and NREAD transactions.
- SRIO is configured for 3.125 Gbps and 5 Gbps speeds with different modes 1x, 2x, and 4x modes.
- The LSUs are set up with default shadow register configuration.

- Doorbell Registers are used to record the time since the MAU (Memory Access Unit) does not have an interrupt. The total time taken for the packet transfer is calculated by using the doorbell time stamps.
- 16 DirectIO transactions are sent to better maximize the throughput between the doorbells.
- Data rate is calculated by dividing the total number of bits by the total time taken.

shows the throughput of a DirectIO operation for an NREAD transaction using a PHY line rate of 3.125 Gbps (SRIO line rate of 2.5Gbps) in 1x, 2x and 4x modes.

For every 256 bytes there is 20 bytes of overhead due to packet header and if the packet size is less than 256 bytes, there will be 18 bytes of overhead.

Table 16 shows the throughput of a DirectIO operation for an NWRITE transaction using a PHY line rate of 3.125 Gbps (SRIO line rate of 2.5 Gbps) in 1x, 2x and 4x modes.

Table 16. DirectIO Write Throughput With 3.125 Gbps PHY

	4x Lanes		2x Lanes		1x Lanes	
Payload Size (Bytes)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)
8	209.72	209.72	209.72	209.72	209.72	209.72
16	419.43	419.43	419.43	419.43	419.43	419.43
32	838.86	838.86	838.86	838.86	838.86	838.86
64	1677.72	1677.72	1677.72	1677.72	1677.72	1677.72
128	3355.44	3355.44	3355.44	3355.44	2118.34	2118.34
256	6710.89	6710.89	4559.03	4559.03	2291.97	2291.97
512	8264.64	8264.64	4583.94	4583.94	2298.25	2298.25
1024	8409.63	8409.63	4596.5	4596.5	2301.93	2301.93
2048	8388.61	8388.61	4619.7	4619.7	2303.24	2303.24
4096	8421.95	8421.95	4614.41	4614.41	2303.9	2303.9
8192	8418.42	8418.42	4618.91	4618.91	2303.24	2303.24

Table 17 shows the throughput of a DirectIO operation for an NREAD transaction using a PHY line rate of 3.125 Gbps (SRIO line rate of 2.5Gbps) in 1x, 2x and 4x modes.

Table 17. DirectIO Read Throughput With 3.125 Gbps PHY1

	4x Lanes		2x Lanes		1x Lanes	
Payload Size (Bytes)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)
8	54.46	52.57	52.32	50.61	49.59	48.19
16	107	103.61	102.27	98.98	96.02	93.23
32	207.09	199.35	196.73	189.84	181.31	176.33
64	386.45	373.82	361.99	350.5	324.8	316.47
128	682.37	662.61	620.77	606.41	533.4	524.07
256	1105.7	1079.15	973.53	953.61	794.38	784.47
512	1956.91	1911.57	1612.68	1587.25	1188.47	1175.97
1024	3169.5	3126.19	2399.6	2369.11	1575.08	1566.26
2048	4608.07	4560.06	3174.5	3149.17	1874.55	1867.42
4096	5956.41	5917.89	3776.17	3760.3	2066.8	2062.46
8192	6976.58	6940.5	4154.92	4144.87	2178.45	2176.03

Table 18 shows the throughput of a DirectIO operation for an NWRITE transaction using a PHY line rate of 5 Gbps (SRIO line rate of 4 Gbps) in 1x, 2x and 4x modes.

Table 18. DirectIO Write Throughput With 5 Gbps PHY

	4x Lanes		2x Lanes		1x Lanes	
Payload Size (Bytes)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)
8	209.72	209.72	209.72	209.72	209.72	209.72
16	419.43	394.2	419.43	419.43	419.43	394.2
32	838.86	838.86	838.86	838.86	838.86	788.4
64	1677.72	1677.72	1677.72	1677.72	1677.72	1677.72
128	3355.44	3355.44	3355.44	3355.44	3355.44	3355.44
256	6710.89	6710.89	6710.89	6710.89	3631.43	3631.43
512	13421.77	13421.77	7262.86	7262.86	3655.17	3655.17
1024	13210.41	13210.41	7374.6	7374.6	3679.21	3679.21
2048	13394.98	13394.98	7358.43	7358.43	3683.93	3683.93
4096	13412.83	13412.83	7388.13	7388.13	3685.61	3685.61
8192	13457.66	13457.66	7389.49	7389.49	3683.93	3683.93

Table 19 shows the throughput of a DirectIO operation for an NREAD transaction using a PHY line rate of 5 Gbps (SRIO line rate of 4 Gbps) in 1x, 2x and 4x modes.

Table 19. DirectIO Read Throughput With 3.125 Gbps PHY

	4x Lanes		2x Lanes		1x Lanes	
Payload Size (Bytes)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)
8	75.4	72.22	72.95	70.09	69.84	65.76
16	148.8	142.21	143.77	137.85	136.18	130.75
32	287.02	274.98	274.26	262.14	256.38	246.14
64	540.97	521.25	511.08	492.29	467.07	450.03
128	957.6	925.21	881.77	851.92	772.91	747.65
256	1572.86	1522.43	1401.22	1369.2	1168.87	1141.83
512	2810.25	2744.36	2361.88	2315.16	1784.18	1751.88
1024	4658.18	4559.03	3596.4	3534.53	2415.73	2388.22
2048	6925.58	6803.87	4861.79	4811.82	2923.28	2903.89
4096	9140.82	9060.6	5890.19	5855.07	3260.35	3248.25
8192	10895.77	10831.29	6554.13	6532.87	3460.41	3453.73

Figure 16 through Figure 19 show DirectIO throughput in NWRITE and NREAD modes, respectively.

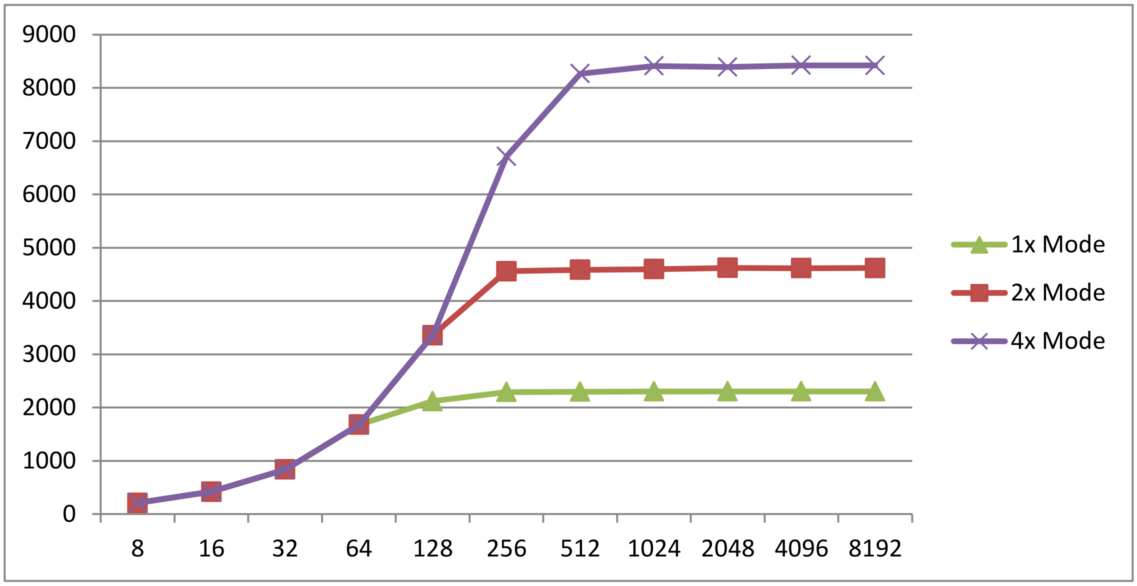


Figure 16. DirectIO NWRITE Throughput With 3 Gbps PHY With Overhead

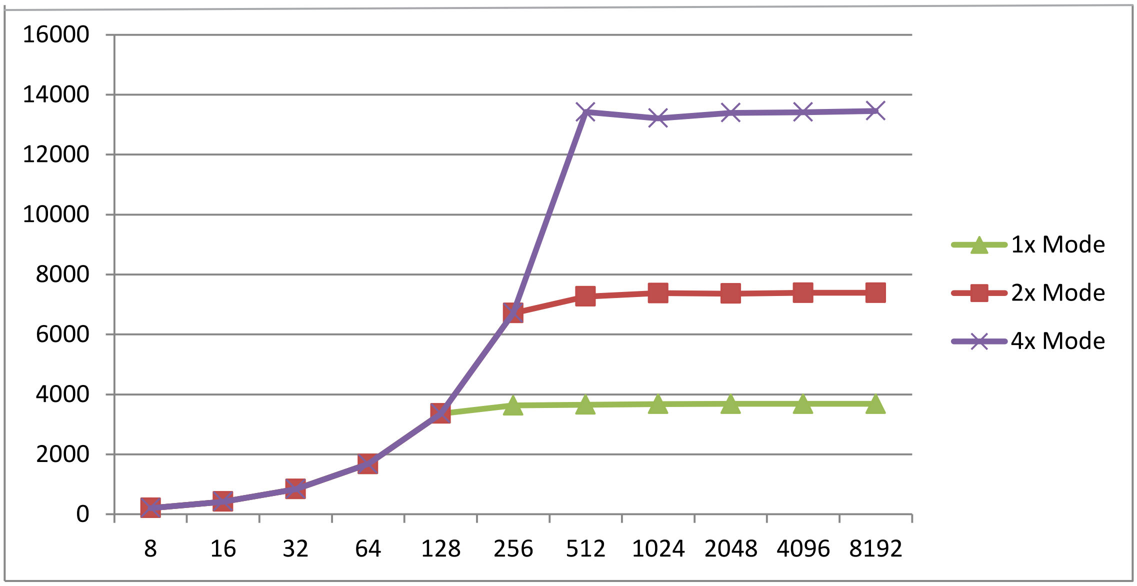


Figure 17. DirectIO NWRITE Throughput With 5 Gbps PHY With Overhead

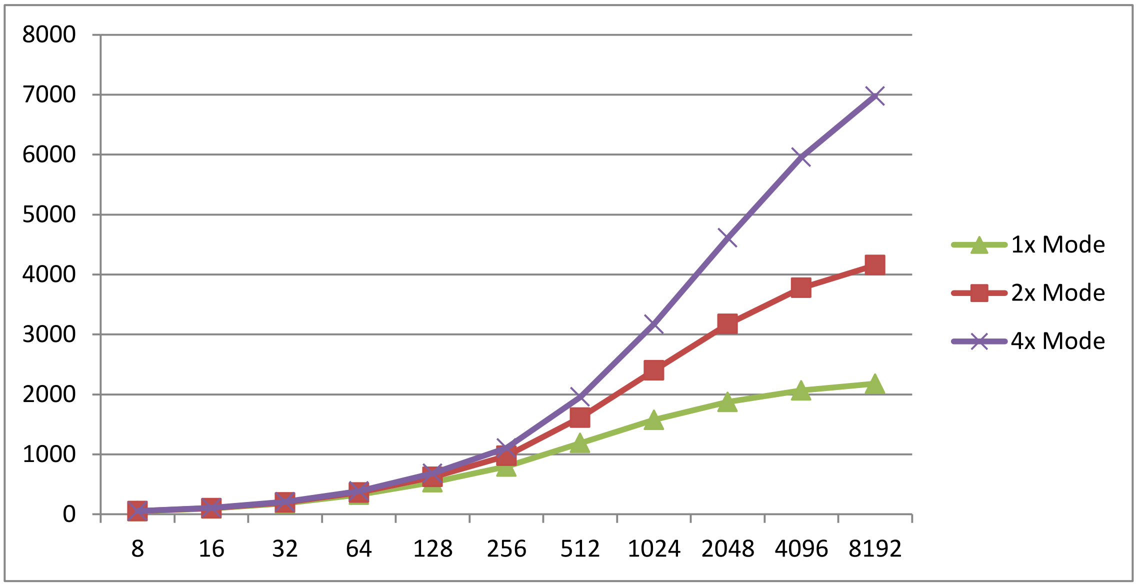


Figure 18. DirectIO NREAD Throughput With 3 Gbps PHY With Overhead

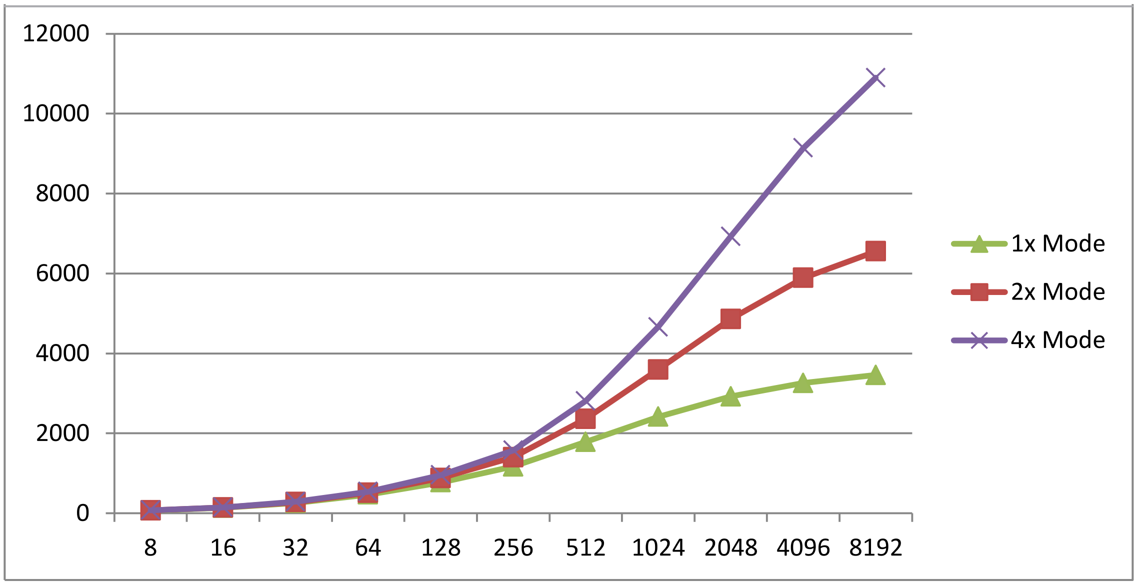


Figure 19. DirectIO NREAD Throughput With 5Gbps PHY With Overhead

7.2 Message Passing Throughput

The Packet DMA module is the incoming and outgoing message passing protocol engine of the SRIO peripheral. With message passing, a mailbox identifier is used instead of a destination address within the RapidIO packet. The mailbox is controlled and mapped to memory by the local (destination) device. For multi-packet messages, four mailbox locations are specified. The Packet DMA transfers messages to the appropriate memory via the DMA bus.

The *ftype* header field of the received RapidIO message packets is decoded by the logical layer of the peripheral. Type 9, Type 11 and Type 13 packets are routed to the Packet DMA.

7.2.1 Type 11 Throughput

The Type 11 message passing tests were measured on a 66AK2H12 operating at 1.2 GHz. L2, MSMC and DDR3 memory endpoints are used for both input and output data. The throughput process includes the following:

1. SRIO is configured for 3.125 Gbps line rate and 5 Gbps line rate with different modes 1x, 2x, and 4x.
2. QMSS Accumulator Firmware is used to monitor the destination queue.
3. All the packets are enqueued by the Host Descriptors. The first few descriptors are programmed for Mailbox 0 and the remaining descriptors are programmed with Mailbox 1 and routed to different RX queues.
4. After enqueuing, the logical transmit channel is enabled to initiate transfer.
5. Multiple packets are sent to get better performance.
6. Time stamps are captured when the interrupts are generated by the QMSS accumulator.
7. 32 transactions are sent for each packet size.
8. 16 Bytes of overhead is added for each packet.

Table 20 and Figure 20 show the throughput of a Type 11 Message Passing operation using a PHY line rate of 3.125 Gbps (SRIO line rate of 2.5 Gbps) in 1x, 2x and 4x modes

Table 20. Type 11 Message Passing Throughput With 3.125 Gbps PHY

Payload Size (Bytes)	4x Lanes		2x Lanes		1x Lanes	
	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)
16	190.65	130.75	190.65	130.75	190.42	130.75
32	381.3	261.49	381.3	261.49	381.3	261.49
64	762.6	522.98	762.6	522.98	762.6	522.98
128	1525.2	1045.96	1525.2	1045.96	1525.2	1055.61
256	3050.4	2091.92	3050.4	2091.92	2304.56	2091.92
512	6100.81	4183.84	4621.82	4183.84	2304.56	2304.56
1024	8423.71	8264.64	4619.7	4619.7	2304.04	2304.04
2048	8423.71	8423.71	4619.7	4619.7	2304.04	2304.04
4096	8421.95	8421.95	4619.17	4619.17	2303.9	2303.9

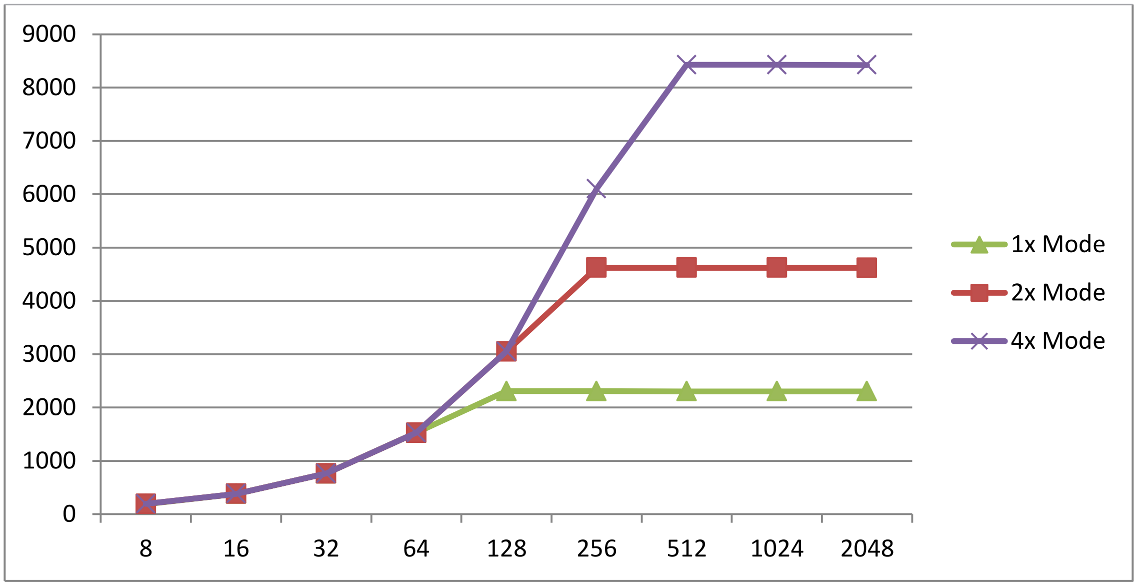


Figure 20. Type 11 Throughput With 3 Gbps PHY With Overhead

Table 21 and Figure 21 show the throughput of a Type 11 Message Passing operation using a PHY line rate of 5 Gbps (SRIO line rate of 4 Gbps) in 1x, 2x and 4x modes.

Table 21. Type 11 Message Passing Throughput With 5 Gbps PHY[1]

	4x Lanes		2x Lanes		1x Lanes	
Payload Size (Bytes)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)	L2 Memory Software Board-to-Board Mode Throughput Value (Mbps)	DDR3 Memory Software Board-to-Board Mode Throughput Value (Mbps)
16	190.65	130.75	190.65	129.13	190.65	130.75
32	381.3	261.49	381.3	258.27	381.3	264.79
64	762.6	522.98	762.6	516.54	762.6	528.25
128	1525.2	1045.96	1525.2	1033.08	1521.51	1059.17
256	3039.35	2091.92	3035.68	2091.92	3050.4	2118.34
512	6100.81	4183.84	6100.81	4183.84	3687.3	3687.3
1024	12201.61	8367.69	7390.84	7390.84	3687.3	3687.3
2048	13475.67	13475.67	7390.84	7390.84	3686.63	3686.63
4096	13475.67	13475.67	7390.84	7390.84	3686.29	3686.29

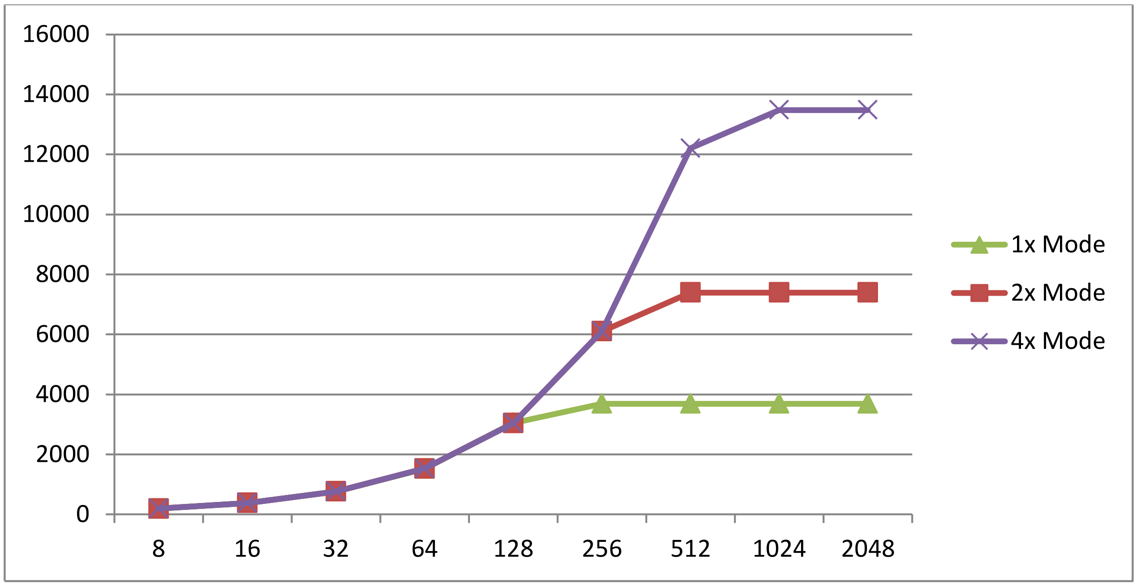


Figure 21. Type 11 Throughput With 5 Gbps PHY With Overhead

8 HyperLink Throughput

HyperLink provides a high-speed, low-latency, and low-pin-count communication interface that bridges TeraNet between two Keystone devices. In bridging TeraNet between two Keystone devices HyperLink uses a simple memory map scheme. The scheme is similar to the one used by PCIe, but it provides more flexible features designed for multicore DSP. The speed of HyperLink on most Keystone DSP's is 10Gbaud. HyperLink uses an encoding scheme for its SerDes lanes that is equivalent to 8b9b. There are 4 SerDes lanes for HyperLink, so the theoretic throughput of a 10Gbaud HyperLink is 10*4*(8/9) = 35.5Gbps = 4.44GB/s.

It is important to note that the 35.5 gigabit per second data rate includes the overhead due to the physical layer, the data link layer (DLL), and the transaction layer packet (TLP). Due to overhead, the raw data that is able to be transferred is less 35.5 gigabits per second.

The remainder of this section discusses:

- Main Factors Affecting HyperLink Throughput
- Measured Throughput

8.1 Main Factors Affecting HyperLink Throughput

This section discusses the main factors that affect the measured throughput of the HyperLink peripheral. To achieve a measurement of the maximum throughput possible using the Keystone HyperLink peripheral, the following factors must be kept in mind:

- Overhead Considerations
- Packet Size Considerations
- EDMA Considerations

8.1.1 Overhead Considerations

The overhead discussed in this section is related to the HyperLink protocol itself, and is not due to any additional use case considerations. As mentioned above, there is a protocol overhead that exists due to HyperLink's 8b9b encoding scheme.

8.1.2 Packet Size Considerations

For each packet transferred over the HyperLink protocol, the overhead (8 bytes) associated with transferring the data must be taken into account. Since this overhead is added on a per packet basis, the amount of overhead can be reduced by making the data payload as large as possible. HyperLink limits the packet size for an individual write data or read data response transaction up to a maximum of 256B (this is alternately stated as maximum burst size in the HyperLink User's Guide) – note that this does not include the control overhead.

The size of write data transactions is limited first by the master that initiates the transaction but may also be limited if it crosses a bridge that fragments the burst size. A CPU will typically have write data limits per transaction of 8B maximum, while the DMAs will typically have write data limits per transaction of 64B or 128B. More information on considerations for EDMA transactions is discussed in Section 8.1.4.

The size of read data response transactions is limited first by the master that initiates the read command itself but may also be limited by the slave entity that is providing the data for the read response. The limits due to the master are the same for reads as for writes. With DMA initiated transactions, the read data response is limited by the slave entity – DDR3, MSMC SRAM, C66x L1/L2. The limits by each of those slave entities are summarized in Table 22.

Table 22. Maximum Read Response Size

Memory	Maximum Read Response
DDR3	64B
MSMC SRAM	32B
C66x L1/L2	16B

8.1.3 Measurement Overhead

The throughput measurement itself has some overhead. Throughput measurement using a small payload size (2KB or less) will be more affected by this overhead. The theoretical calculation is targeted for when the system reaches steady state, which can take a while to achieve. So the throughput measurement should use larger payload sizes (larger than 2KB) to get a good measurement so that the measurement overhead becomes an insignificant portion of the payload transfer time. The throughput in Section 8.2 is calculated by trending the average throughput over multiple data payload sizes.

8.1.4 EDMA Considerations

When using EDMA to transmit data to the Keystone HyperLink, or when using EDMA to receive data from the Keystone HyperLink, the transfer controller (TC) that is used should be taken into consideration. EDMA TCs fragment read and write commands into data-burst-sized (DBS) chunks. Not all transfer controllers use the same data burst size (DBS), and the size of the data burst can have an impact on the throughput of the HyperLink peripheral. EDMA TCs support a DBS of 64B or 128B. Since the HyperLink peripheral supports packet sizes of up to 256B (not including the control overhead) an EDMA TC that supports the larger of the DBS sizes (128B) should be chosen for writes, if available. Overhead will be introduced every 64-bytes or 128-bytes for TCs with DBS=64B or 128B, respectively. For reads the selection of the EDMA TC is less critical because the read response data size will be limited by the slave entity as listed in Table 22.

The placement of TCs on the interconnect also impacts throughput. EDMA's connected to wider interconnects and placed "closer" to the HyperLink (like EDMA CC0 and EDMA CC4) will generally result in higher throughput than EDMA's that are "farther" and connected to narrower interconnects (like EDMA CC1, EDMA CC2 and EDMA CC3).

The bottleneck of the throughput is almost always the HyperLink and not the EDMA transfer controller.

8.2 Measured Throughput

The measurements mentioned in this section were collected on a 66AK2H12 device but can be used for any device from the Keystone-II family that has HyperLink.

Throughput data for the HyperLink peripheral is captured in Table 23 and Table 24. For all throughput tests, both links of the HyperLink module on one DUT were connected directly with links on the HyperLink module on a second DUT.

For this test, both DUT HyperLink peripherals are configured a line rate 6.25 Gbaud with x4 mode (4 HyperLink lanes are enabled). HyperLink on the first DUT performs write/read transactions with the second DUT's slave memory (L2, MSMC SRAM, and DDR3) via HyperLink port. Data was collected for 8KB, 16KB, and 32KB data payloads. The difference in measured throughput between different data transfer sizes was not significant; the throughput in the following tables was obtained by trending the average throughput over these three data payloads.

Throughput, while writing over HyperLink, is much better than reading over HyperLink. The reason lies in how the payloads are fragmented for writes v/s reads and is described below.

Write throughput: During writes, HyperLink fragments a payload size larger than 256B at 256B address boundary. Overhead of 8 bytes is thus added for each 256B packet. There is also the overhead associated with 8b/9b encoding. Therefore, the maximum write throughput for (6.25 Gbaud line rate) is 25Gbps*(8/9)*256/ (256+8) = 21.54Gbps. However, there are other factors internal to the architecture – including the choice of EDMA channel controller – that introduce additional overhead and further reduce the maximum achievable write throughput below 21.54Gbps as can be seen in Table 23. Table 24 shows the write throughput measured on 66AK2H12 silicon.

Table 23. Maximum Achievable Hyperlink Write Throughput (6.25Gbaud x 4 lanes)

Memory	EDMA CC0 and CC4 (Gbps)	EDMA CC1, CC2 and CC3 (Gbps)
Write to L2	21.05	18.71
Write to MSMC	21.05	18.71
Write to DDR3	21.05	18.71

Table 24. Measured Hyperlink Write Throughput (Gbps and % of max achievable throughput)

Memory	EDMA CC0 and CC4 (Gbps)	EDMA CC1, CC2 and CC3 (Gbps)
Write to L2	20.14 (96%)	18.05 (96%)
Write to MSMC	20.00 (95%)	18.10 (97%)
Write to DDR3	20.19 (96%)	18.28 (98%)

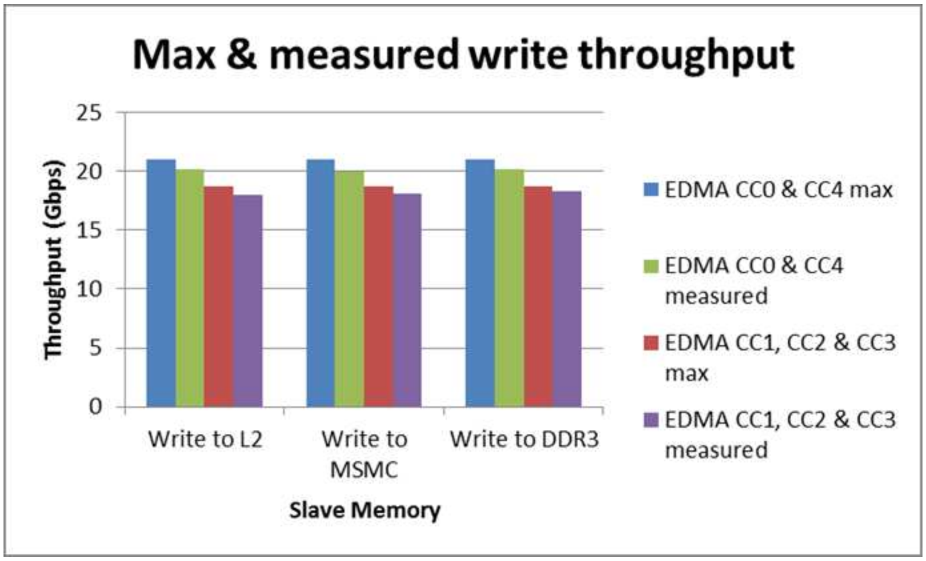


Figure 22. Maximum Achievable and Measured Write Throughput

Read throughput: Read throughput depends on the read response size of the memory end point accessed by HyperLink, as described in Section 8.1.2. The MSMC SRAM returns reads as 32B bursts, the local L2s return reads in 16B bursts and the DDR3 returns reads in 64B bursts. The HyperLink returns each read burst individually, therefore, the read from MSMC SRAM can reach 25Gbps*(8/9)*32/(32+8)= 17.77Gbps, the read from a local L2 can reach 25Gbps*(8/9)*16/(16+8)= 14.81Gbps and read from DDR3 can reach 25Gbps*(8/9)*64/(64+8)= 19.54Gbps Gbps. Similar to write throughput, additional overhead further reduces the maximum achievable read throughput as well, as can be seen in Table 25. The read throughput depends on the read response size and is not impacted by the selection of EDMA channel controller. Table 26 shows the read throughput measured on 66AK2H12 silicon.

Table 25. Maximum Achievable Hyperlink Read Throughput (6.25Gbaud x 4 lanes)

Memory	EDMA CC0 and CC4 (Gbps)	EDMA CC1, CC2 and CC3 (Gbps)
Write to L2	12.69	12.69
Write to MSMC	16.16	16.16
Write to DDR3	18.71	18.71

Table 26. Measured Hyperlink Read Throughput (Gbps and % of max achievable throughput)

Memory	EDMA CC0 and CC4 (Gbps)	EDMA CC1, CC2 and CC3 (Gbps)
Write to L2	11.31 (89%)	8.91 (70%)
Write to MSMC	15.08 (93%)	10.02 (62%)
Write to DDR3	13.39 (72%)	8.63 (46%)

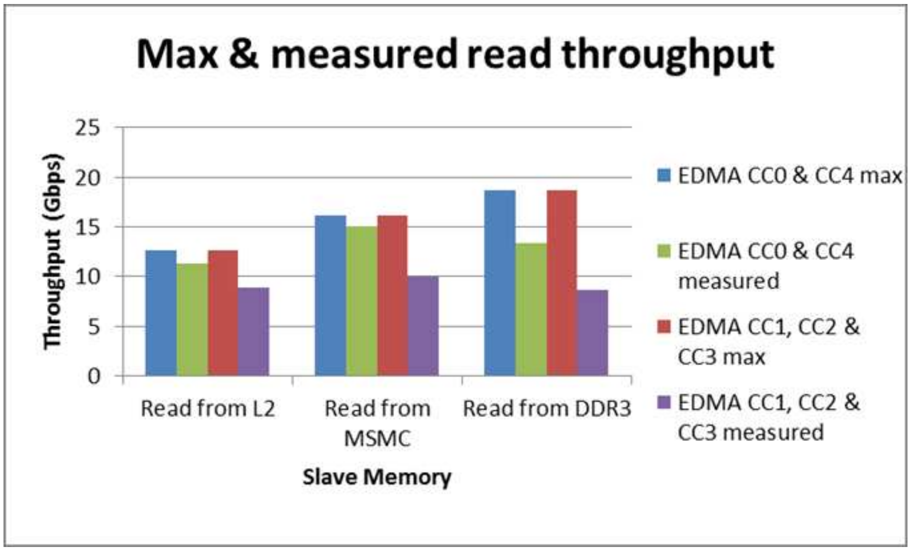


Figure 23. Maximum and Measured Read Throughput

9 10 Gigabit Ethernet Throughput

This section describes the 10GbE throughput measured on 66AK2H12 silicon. The results apply equally to all other Keystone-II devices that support a 10GbE interface.

The tests were run in a bare metal environment. Throughput was measured for multiple packet sizes. The Packet DMA descriptors were placed in MSMC SRAM and data buffers were placed in DDR3 memory.

An RTM bridge from Advantech is used to connect two K2H EVMs over their 160-pin ZDplus RTM connectors. Figure 24 shows the hardware setup. Arrows show the dataflow. Throughput is measured only on a single lane as indicated by the grey box. EVM1 is used for transmit and EVM2 is used for receive.

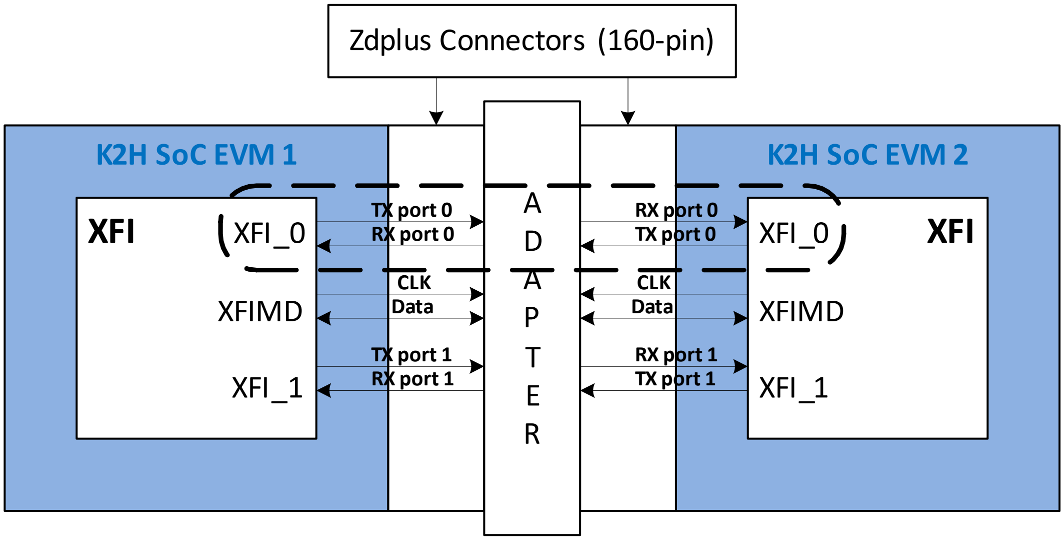
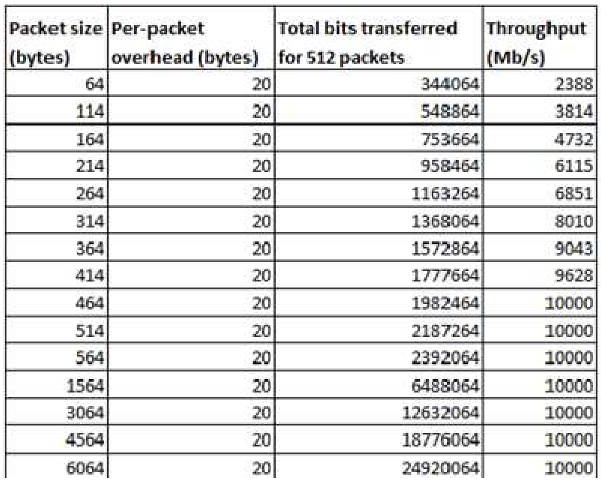


Figure 24. Hardware Setup for Throughput Measurement

EVM1 (transmit side): the 10GbE module’s Packet DMA transmit channels are disabled. 512 packets are then placed onto the disabled transmit channel's queues. Once the 512 packets are waiting at the transmit channels, the channels are enabled.

EVM2 (receive side): On the receive side, the receive flows of the 10GbE Packet DMA are configured to use the same destination queue regardless of which external port the packets entered the switch. The throughput program polls on this destination queue waiting for the first packet to arrive. Once the first packet arrives, a timestamp is taken. After receiving all 512 packets, a second timestamp is taken. The difference in time and the number of bytes received is used to calculate the throughput. The maximum achievable throughput on a single lane is 10Gbps.

The throughput calculations include the standard per-packet overhead of 20 bytes that includes Inter Packet Gap (IPG), Preamble and Start of Frame Delimiter (SOD). Note that the Packet DMA also adds some overhead when it fetches and updates a descriptor for each arriving packet. For more details on Packet DMA behavior, see the *KeyStone Architecture Multicore Navigator User's Guide* (SPRUGR9). This overhead disproportionally impacts the smaller packet sizes, resulting in lower measured throughput. The significance of the overhead decreases as packet sizes are progressively increased, causing the throughput to rise as well. For the larger packet sizes, the packet size is large enough to hide this overhead within the receive time of the next packet, causing the measured throughput to ultimately flatten out at the maximum line rate of 10Gbps. The packet selection is finer grained (50 bytes) for the smaller packets and coarser (1500 bytes) for the larger packets.



Packet size (bytes)	Per-packet overhead (bytes)	Total bits transferred for 512 packets	Throughput (Mb/s)
64	20	344064	2388
114	20	548864	3814
164	20	753664	4732
214	20	958464	6115
264	20	1163264	6851
314	20	1368064	8010
364	20	1572864	9043
414	20	1777664	9628
464	20	1982464	10000
514	20	2187264	10000
564	20	2392064	10000
1564	20	6488064	10000
3064	20	12632064	10000
4564	20	18776064	10000
6064	20	24920064	10000

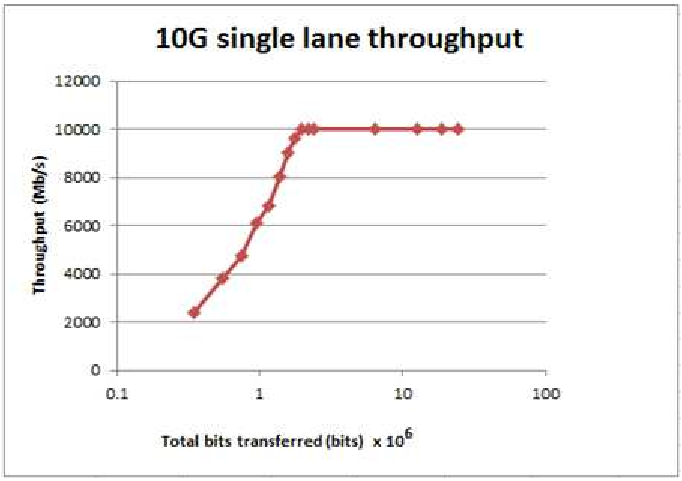


Figure 25. Measured Throughput

10 References

- *TMS320C66x DSP Cache User's Guide* (SPRUGY8)
- *KeyStone Architecture Multicore Navigator User's Guide* (SPRUGR9)
- *KeyStone Architecture Peripheral Component Interconnect Express (PCIe) User's Guide* (SPRUGS6)
- *Keystone II Architecture Multicore Shared Memory Controller (MSMC) User’s Guide* (SPRUHJ6)

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from A Revision (June 2012) to B Revision	Page
• Updates were made to Section 2	3
• Added new information for 'SRIO Throughput' in Section 7	28
• Added new information in Section 7.1	28
• Added new information in Section 7.2.1.	33

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

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

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

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

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

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

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

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

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

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