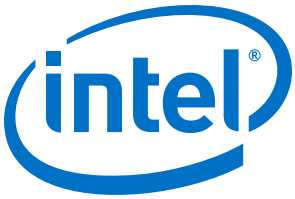


AN 881: PCI Express* Gen3 x16 Avalon®-MM DMA with External and HBM2 Memories Reference Design

Subscribe
Send Feedback

AN-881 | 2019.07.19
Latest document on the web: PDF | HTML

Contents

1. Introduction

This document covers a reference design using the PCI Express* Avalon® Memory-Mapped (Avalon-MM) Direct Memory Access (DMA) with Memory IP Interfaces. This reference design demonstrates the performance of the Avalon-MM Intel® Stratix® 10 Hard IP+ for PCI Express, a high-performance DMA controller with two types of memory solutions: external (DDR4) and HBM2 memories.

The reference design includes a Linux* software driver to set up the DMA transfers with high-throughput data movers for DMA support. The Read Data Mover moves data from the system memory to the external or HBM2 memory in Avalon-MM space . The Write Data Mover moves data from the external or HBM2 memory in the application logic to the system memory in PCIe* space. This reference design allows you to evaluate the performance of the Avalon-MM Intel Stratix 10 Hard IP+ for PCI Express while using the Avalon-MM interface with high-performance DMA with different memory IPs.

Figure 1. Avalon-MM DMA and Address Mapping Block Diagram

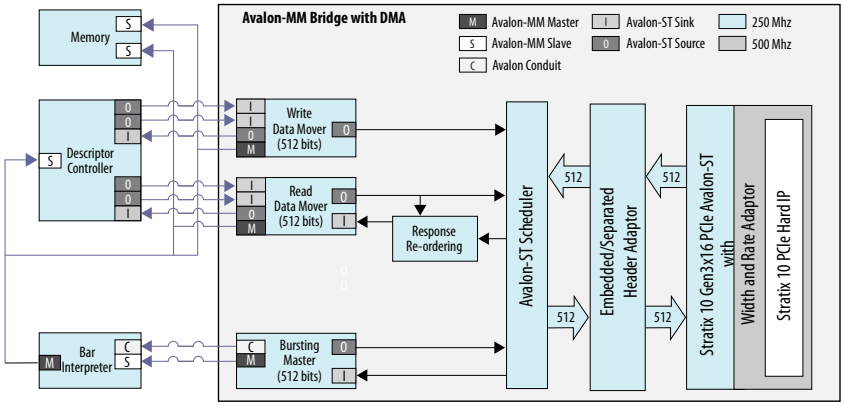


Table 1. Reference Design Information

Reference Design	Hardware	Throughput (GB/s)		Gate Counts	Design Link
		Read	Write		
Avalon-MM Intel Stratix 10 MX Hard IP+ DMA with HBM2 and DDR4	Intel Stratix 10 MX FPGA Development Kit	13.95	13.01	77K ALMs 85K ALUTs 713 M20Ks	PCI Express Gen3 x16 AVMM DMA with HBM2 and DDR4 Reference Design

Note: To download this reference designs, first make sure that you have access to the Intel Design Store by logging into Design Store. You can then click on the link provided in the table above to download the design.

Note: Although the theoretical maximum throughput for either Read or Write operations is 16 GB/s, the real throughput will be less than that (as shown in the table above) due to the overhead inherent in the PCI Express protocol.

For Intel Arria® 10, Intel Cyclone® 10 GX or Intel Stratix 10 SX, GX or TX devices and Avalon-MM DMA configurations up to Gen3 x8, refer to:

- Intel Stratix 10 Avalon-MM Interface for PCI Express Solutions User Guide
- AN 829: PCI Express Avalon-MM DMA Reference Design
- Intel Arria 10 or Intel Cyclone 10 GX Avalon-MM DMA Interface for PCI Express Solutions User Guide

Related Information

- Avalon-MM Intel Stratix 10 Hard IP+ for PCI Express Solutions User Guide
- PCI Express Base Specification Revision 3.0
- High Bandwidth Memory (HBM2) Interface Intel FPGA IP User Guide
- External Memory Interfaces Intel Stratix 10 FPGA IP User Guide
- Intel Stratix 10 MX FPGA Development Kit

1.1. PCI Express Gen3 x16 Avalon-MM DMA with External and HBM2 Memory Hardware and Software Requirements

Hardware Requirements

The reference design runs on the following development kits:

- Intel Stratix 10 FPGA MX development kit (using **UDIMM** DDR4 and HBM2 memories). This development kit uses a 1SM21BHU2F53E2VG Intel Stratix 10 device.

The reference design requires two computers:

- A computer with a PCIe Gen3 x16 slot running Linux. This computer is computer number 1.
- A second computer with the Intel Quartus® Prime Pro Edition software version 19.1 installed. This computer downloads the FPGA SRAM Object File (.sof) to the FPGA on the development kit. This computer is computer number 2.

Software Requirements

- The reference design software is installed on computer number 1. The reference design is available in the Intel FPGA Design Store. The Intel Quartus Prime Pro Edition Platform Archive File (.par) includes the recommended synthesis, fitter, and timing analysis settings for the parameters specified in the reference design.
- The Intel Quartus Prime Pro Edition software is installed on computer number 2. You can download this software from the Intel Quartus Prime Pro Edition Software Features/Download web page.
- The Linux driver is configured specifically for this reference design.

Note: The driver was developed and tested on CentOS 7.0, 64-bit with 3.10.514 kernel compiled for the x86_64 architecture.

Related Information

- Intel Stratix 10 PCI Express Gen3 x16 Avalon-MM DMA with External and HBM2 Memories Reference Design

To download the reference design and the design software from the Design Store, go to:

- Intel Quartus Prime Pro Edition Download Center: http://fpgasoftware.intel.com/?edition=pro

1.2. Avalon-MM Bridge with DMA Module Description

Refer to the *Avalon-MM Intel Stratix 10 Hard IP+ for PCI Express Solutions User Guide*.

Related Information

Avalon-MM Intel Stratix 10 Hard IP+ for PCI Express Solutions User Guide

1.3. DMA Procedure Steps

Software running on the host completes the following steps to initiate the DMA and verify the results.

Refer to *AN 829: PCI Express Avalon-MM DMA Reference Design* for the DMA procedure.

Related Information

https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/an/an829.pdf

1.4. Setting Up the Hardware

1. Power down computer number 1.
2. Plug the FPGA MX development kit card into a PCIe slot that supports Gen3 x16.
3. For the Intel Stratix 10 FPGA MX development kit, connectors J26 and J27 power the card. After inserting the card into an available PCIe slot, connect 2x4- and 2x3-pin PCIe power cables from the power supply of computer number 1 to the J26 and J27 connectors of the PCIe card, respectively.
4. Connect a USB cable from computer number 2 to the FPGA MX development kit. The development kit includes an on-board Intel FPGA Download Cable for FPGA programming.
5. To power up the FPGA MX development kit via the PCIe slot, power on computer number 1. Alternatively, you can power up the FPGA MX development kit using the external power adapter that ships with the kit.
6. On computer number 2, bring up the Intel Quartus Prime programmer and configure the FPGA through an Intel FPGA Download Cable.

   Note: You must reconfigure the FPGA whenever the FPGA development kit loses power.
7. To force system enumeration to discover the PCIe device, restart computer number 1.

1.5. Installing the DMA Test Driver and Running the Linux DMA Software

For instructions on how to install the DMA test driver and run the Linux DMA application, refer to sections 2.6 and 2.7 in the User Guide for the Avalon-MM Intel Stratix 10 Hard IP+ for PCI Express (Avalon-MM Intel Stratix 10 Hard IP+ for PCI Express Solutions User Guide).

Note: After downloading the reference design from the Design Store and decompressing the archived project (from the .qar file) using Intel Quartus Prime 19.1, you will find a few folders generated in your project directory. You will also have a `software.zip` folder with the software driver and tool. Make sure to unzip this folder in Windows. You may run into driver installation issues if you unzip this folder in Linux.

Figure 2. Link Test GUI

```
************************************************************
Intel FPGA PCIe Link Test
Version 2.0
0: Automatically select a device
1: Manually select a device
************************************************************
> 1
Enter bus number, in hex:
> 2
Enter device number, in hex:
> 0
Enter function number, in hex:
> 0
BDF is 0x200
B:D.F, in hex, is 2:0.0
Enter BAR number (-1 for none, 2 for HBM2 and 4 for DDR4)) :
> 4
Opened a handle to BAR 0x4 of a device with BDF 0x200

************************************************************
 0: Link test - 100 writes and reads
 1: Write memory space
 2: Read memory space
 3: Write configuration space
 4: Read configuration space
 5: Change BAR (HBM2 = 2 and DDR4 = 4)
 6: Change device
 7: Enable SRIOV
 8: Do a link test for every enabled virtual function
    belonging to the current device
 9: Perform DMA
10: Quit program
************************************************************
>
```

Figure 3. Link Test Pass Result

```
 0: Link test - 100 writes and reads
 1: Write memory space
 2: Read memory space
 3: Write configuration space
 4: Read configuration space
 5: Change BAR (HBM2 = 2 and DDR4 = 4)
 6: Change device
 7: Enable SRIOV
 8: Do a link test for every enabled virtual function
    belonging to the current device
 9: Perform DMA
10: Quit program
*******************************************************
> 0
Doing 100 writes and 100 reads..
Number of write errors:       0
Number of read errors:        0
Number of dword mismatches:   0

*******************************************************
 0: Link test - 100 writes and reads
 1: Write memory space
 2: Read memory space
 3: Write configuration space
 4: Read configuration space
 5: Change BAR (HBM2 = 2 and DDR4 = 4)
 6: Change device
 7: Enable SRIOV
 8: Do a link test for every enabled virtual function
    belonging to the current device
 9: Perform DMA
10: Quit program
*******************************************************
>
```

Figure 4. DMA GUI

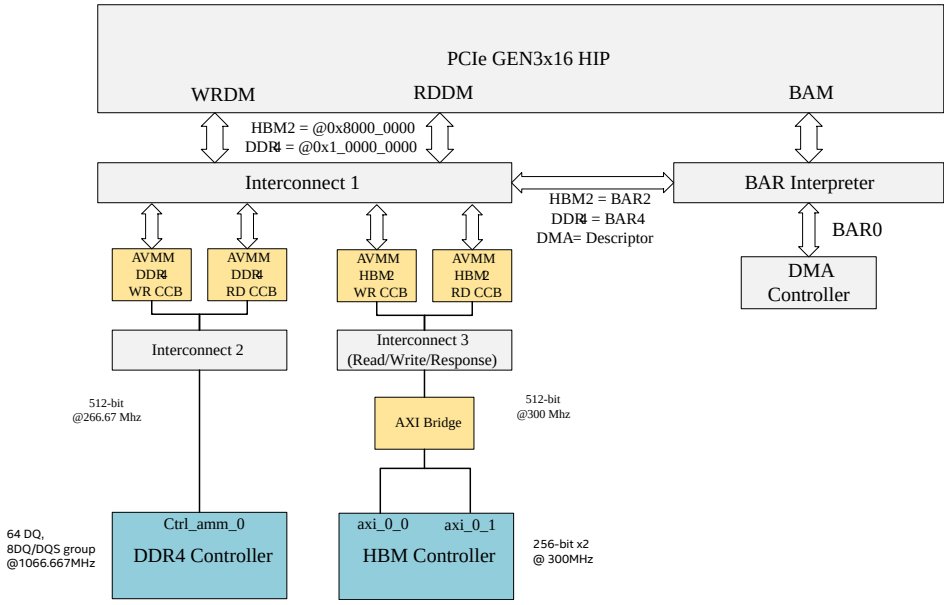
```
 1: Write memory space
 2: Read memory space
 3: Write configuration space
 4: Read configuration space
 5: Change BAR (HBM2 = 2 and DDR4 = 4)
 6: Change device
 7: Enable SRIOV
 8: Do a link test for every enabled virtual function
    belonging to the current device
 9: Perform DMA
10: Quit program
*******************************************************
> 9

*******************************************************
Current DMA configurations
    Memory Interface          ? HBM2
    Run Read  (card->system)  ? 1
    Run Write (system->card)  ? 1
    Run Simultaneous          ? 1
    Number of dwords/desc     : 16384
    Number of descriptors     : 128
    Total length of transfer  : 8192 KiB
*******************************************************
 0: Select DMA interface (HBM2 or DDR4)
 1: Run DMA
 2: Toggle read DMA
 3: Toggle write DMA
 4: Toggle simultaneous DMA
 5: Set the number of dwords per descriptor
 6: Set the number of descriptors per DMA
 7: Return to main menu
*******************************************************
> 
```

2. Reference Design Description

This application note consists of a reference design using the Avalon-MM Intel Stratix 10 MX Hard IP+ DMA with external DDR4 and HBM2 memories.

Figure 5. Gen3 x16 DMA with DDR4 and HBM2 Memories Reference Design



This Gen3 x16 DMA with DDR4 and HBM2 Platform Designer system (captured in the file `g3x16_hbm2_ddr4.qsys`) instantiates four Avalon-MM clock-crossing bridges and an AXI bridge IP core between the PCIe Avalon-MM Masters, DDR4 and HBMC AXI Slave. The purpose of those IPs is to perform the following functions:

- PCIe Hard IP and HBMC clock domain crossing
- PCIe Hard IP and DDR4 clock domain crossing
- Burst length adaptation
- Exporting the AXI Master interface
- Controlling the Read/Write Response FIFO depth

PCIe Hard IP and Memory clock domain crossing

The Gen3 x16 IP user interface is 512-bit @ 250 MHz. The 250 MHz is the frequency of the `coreclkout_hip` generated by the PCIe Hard IP. The HBM Controller AXI interface in the design is 256-bit @ 300 MHz. The HBM Controller core clock is generated by an IOPLL. Two Avalon-MM clock-crossing bridges are used to handle the clock crossing.

The DDR4 Controller interface in the design uses 512-bit @ 266.67 MHz. Two Avalon-MM clock-crossing bridges are used to handle the clock crossing.

Burst length adaptation

The Gen3 x16 IP Write Data Mover (WRDM) and Read Data Mover (RDDM) Avalon-MM interfaces are bursting masters that issue Read/Write transactions in burst mode (the maximum burst count supported is 8). However, the HBM Controller AXI4 slave only supports single-burst transfers (burst length of 1). To resolve this, the maximum burst size in the Avalon-MM clock crossing bridges is set to 1.

Exporting the AXI Master interface

The design uses an AXI bridge to export the AXI Master interface from the Platform Designer system. The exported AXI Master interface is connected externally to the HBMC AXI Slave interfaces.

The AXI Bridge Read/Write address drives both HBMC AXI slaves.

The AXI Bridge Read/Write 512-bit data bus is split into two 256-bit data busses.

For a picture of how the AXI Bridge is incorporated into the reference design, refer to Figure 5 on page 9.

Controlling the Read/Write Response FIFO depth

The AXI Bridge Read/Write Acceptance Capability parameter setting dictates the Interconnect Read/Write Response FIFO depth generated by Platform Designer in the `altera_merlin_axi_slave_ni` module. The Response FIFO depth affects the Avalon-MM transaction performance.

If the Read/Write Response FIFO depth is not deep enough and the FIFO becomes full, it creates backpressure, impacting the throughput.

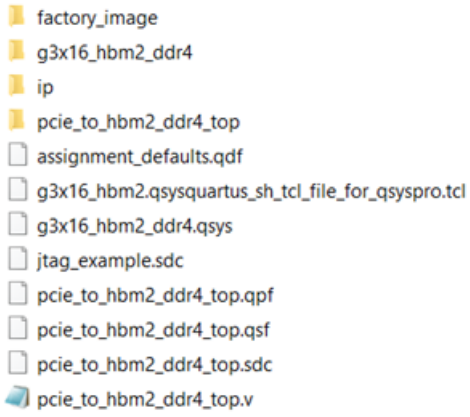
The default Read/Write Acceptance Capability parameter value is set to 16. Intel Quartus Prime 19.1 allows up to 32. In this design, the Read/Write Response FIFO depth generated by Platform Designer is manually changed to 64 in the `altera_merlin_axi_slave_ni` module in order to support Gen3 x16 throughput.

In the 19.3 Intel Quartus Prime release, the AXI Bridge will support a higher maximum value.

2.1. Project Hierarchy

The reference design uses the following directory structure:

Figure 6. Directory Structure



- pcie_to_hbm2_ddr4_top.v: The top-level module.

2.2. Parameter Settings for PCI Express Hard IP Variations

This reference design supports a 512-byte maximum payload size. The following tables list the values for all the parameters.

Table 2. System Settings

Parameter	Value
Number of lanes	Intel Stratix 10 MX: 16
Lane rate	Intel Stratix 10 Gen3: 8 Gbps
Hard IP Mode	By default, the Hard IP mode is set to Gen3 x16, with a 512-bit interface to the Application Layer running at 250 MHz.

Table 3. Base Address Register (BAR) Settings

Parameter	Value	BAR Size
BAR0	64-bit prefetchable memory	DMA: 16 bits
BAR1	Disabled	
BAR2	64-bit prefetchable memory	HBM2: 28 bits
BAR3	Disabled	
BAR4	64-bit prefetchable memory	DDR4: 30 bits
BAR5	Disabled	

Table 4. Device Identification Register Settings

Parameter	Value
Vendor ID	0x00001172
Device ID	0x0000E003
Revision ID	0x00000001

continued...

Parameter	Value
Class Code	0x00000000
Subsystem Vendor ID	0x00000000
Subsystem Device ID	0x00000000

Table 5. PCI Express/PCI* Capabilities

Parameter	Value
Maximum payload size	512 bytes
Completion timeout range	None
Implement completion timeout	Disabled

Table 6. Error Reporting Settings

Parameter	Value
Advanced Error Reporting (AER)	Enabled
ECRC checking	Disabled
ECRC generation	Disabled

Table 7. Link Settings

Parameter	Value
Link port number	1
Slot clock configuration	Enabled

Table 8. Message Signaled Interrupts (MSI) and MSI-X Settings

Parameter	Value
Number of MSI messages requested	4
Implement MSI-X	Disabled
Table size	0
Table offset	0x0000000000000000
Table BAR indicator	0
Pending bit array (PBA) offset	0x0000000000000000
PBA BAR indicator	0

Table 9. Power Management

Parameter	Value
Endpoint L0s acceptable latency	Maximum of 64 ns
Endpoint L1 acceptable latency	Maximum of 1 us

Table 10. PCIe Address Space Setting

Parameter	Value
Address width of accessible PCIe memory space	40

Table 11. DDR4 Memory

Parameter	Value
Memory format	UDIM

2.3. PCIe Avalon-MM DMA Reference Design with External and HBM2 Memories Platform Designer System

The following image shows the modules in the Platform Designer system for this reference design.

Figure 7. Avalon-MM Intel Stratix 10 Hard IP+ for PCI Express with External and HBM2 Memories Platform Designer System

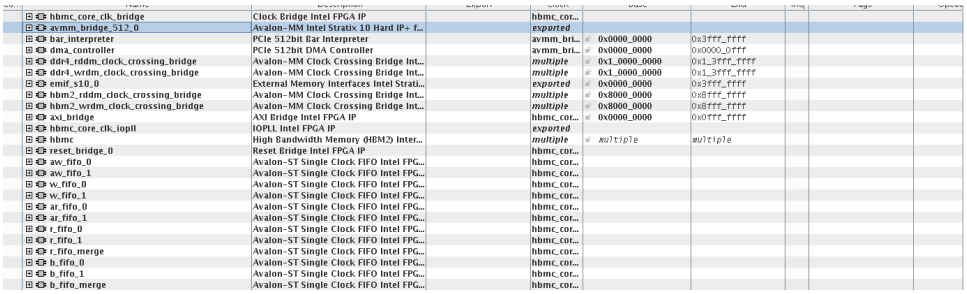


Table 12. Address Mapping

Interface	Base and End Addresses	Bursting Avalon Master (BAM) BAR
Intel DMA Controller	0x0000_0000 - 0x0000_0fff	BAR0
HBM2 Memory Controller	0x8000_0000 - 0x8fff_ffff	BAR2
Intel DDR4 Controller	0x1_0000_0000 - 0x1_3fff_ffff	BAR4

Table 13. Platform Designer Port Descriptions

Port	Function	Description
RDDM	Read Data Mover	This interface transfers DMA data from the PCIe system memory to the memory in Avalon-MM address space.
WRDM	Write Data Mover	This interface transfers DMA data from the memory in Avalon-MM address space to the PCIe system memory.
BAM	Bursting Avalon-MM Master	This interface provides host access to the registers and memory in Avalon-MM address space. The Bursting Avalon-MM Master module converts PCIe Memory Reads and Writes to Avalon-MM Reads and Writes.
Intel DDR4 Controller	DDR4 Controller	This is a single-port DDR4 controller with 64 DQ width and 8 DQ per DQS group.
HBM2 Memory Controller	HBM Controller	This is a single channel with 2 pseudo channel HBM Controllers. The user interface to the HBM2 Controller uses

continued...

Port	Function	Description
		the AXI4 protocol. Each Controller has one AXI4 interface per pseudo channel or 2 AXI4 interfaces per channel.

2.4. Installing the DMA Test Driver and Running the Linux DMA Software

For instructions on how to install the DMA test driver and run the Linux DMA application, refer to sections 2.6 and 2.7 in the User Guide for the Avalon-MM Intel Stratix 10 Hard IP+ for PCI Express (Avalon-MM Intel Stratix 10 Hard IP+ for PCI Express Solutions User Guide).

Figure 8. Link Test GUI

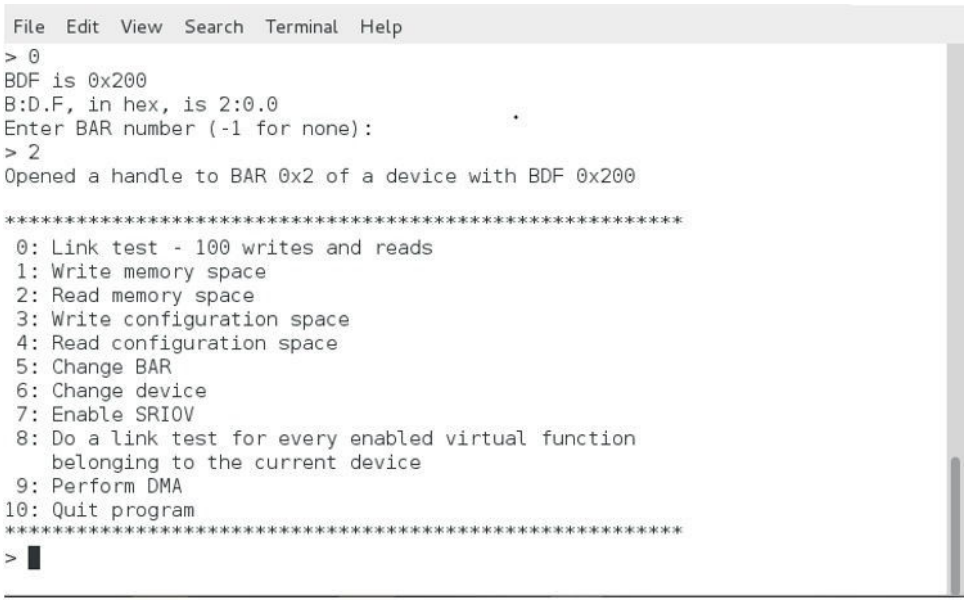


Figure 9. Link Test Pass Result

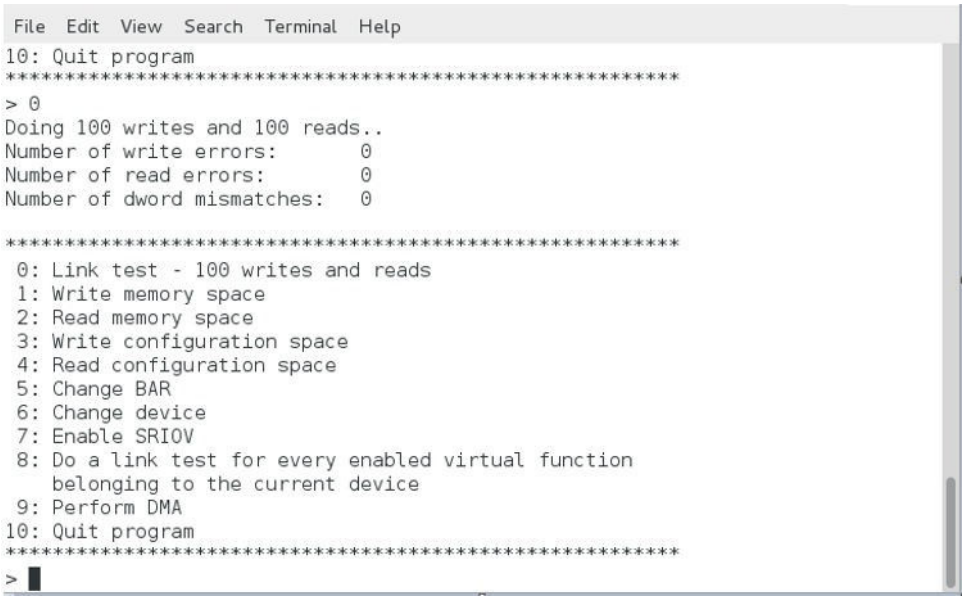
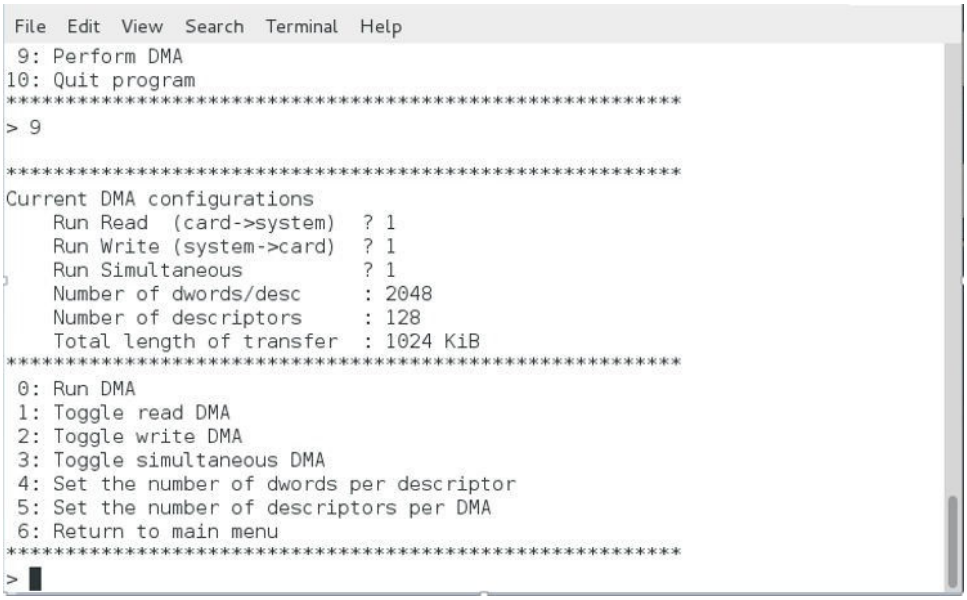
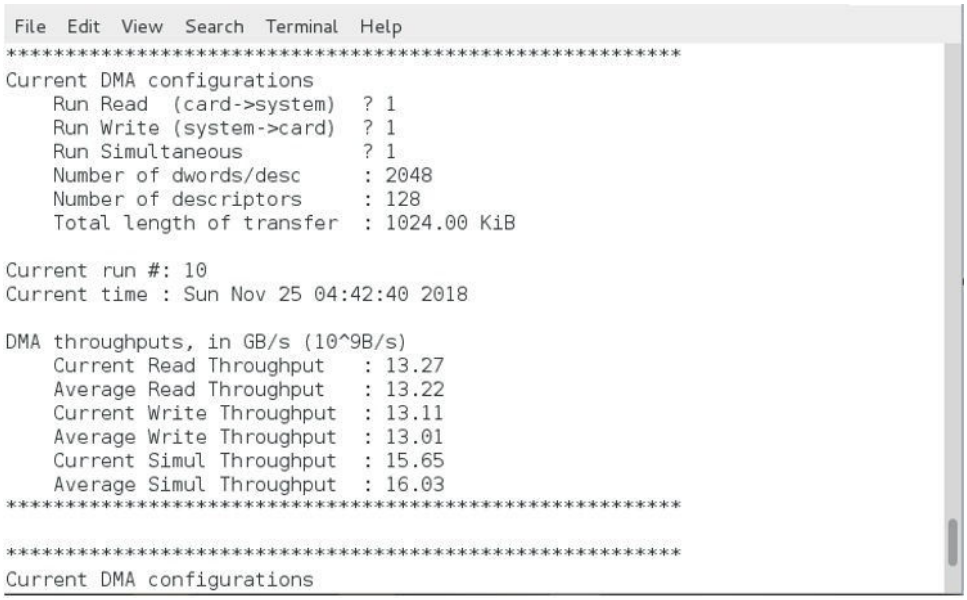


Figure 10. DMA GUI



2.5. Intel Stratix 10 MX DMA Memory Throughput

Figure 11. Intel Stratix 10 MX DMA Memory Throughput



3. Understanding PCI Express throughput

3.1. Throughput for Posted Writes

The theoretical maximum throughput calculation uses the following formula:

```
Throughput  = payload size / (payload size + overhead) * link data rate
```

3.1.1. Specifying the Maximum Payload Size

The `Device Control` register, bits [7:5], specifies the maximum TLP payload size of the current system. The `Maximum Payload Size` field of the `Device Capabilities` register, bits [2:0], specifies the maximum permissible value for the payload. You specify this read-only parameter, called **Maximum Payload Size**, using the parameter editor. After determining the maximum TLP payload for the current system, software records that value in the `Device Control` register. This value must be less than the maximum payload specified in the `Maximum Payload Size` field of the `Device Capabilities` register.

Understanding Flow Control for PCI Express

Flow control guarantees that a TLP is not transmitted unless the receiver has enough buffer space to accept the TLP. There are separate credits for headers and payload data. A device needs sufficient header and payload credits before sending a TLP. When the Application Layer in the completer accepts the TLP, it frees up the RX buffer space in the completer's Transaction Layer. The completer sends a flow control update packet (FC Update DLLP) to replenish the consumed credits to the initiator. When a device consumes all its credits, the rate of FC Update DLLPs to replenish header and payload credits limits throughput. The flow control updates depend on the maximum payload size and the latencies of two connected devices.

3.2. Throughput for Reads

PCI Express uses a split transaction model for reads. The read transaction includes the following steps:

1. The requester sends a Memory Read Request.
2. The completer sends out the ACK DLLP to acknowledge the Memory Read Request.
3. The completer returns a Completion with Data. The completer can split the Completion into multiple completion packets.

Read throughput is typically lower than write throughput because reads require two transactions instead of a single write for the same amount of data. The read throughput also depends on the round trip delay between the time when the

Application Layer issues a Memory Read Request and the time when the requested data returns. To maximize the throughput, the application must issue enough outstanding read requests to cover this delay.

To maintain maximum throughput for the completion data packets, the requester must optimize the following settings:

- The number of completions in the RX buffer
- The rate at which the Application Layer issues read requests and processes the completion data

Read Request Size

Another factor that affects throughput is the read request size. If a requester requires 4 KB of data, the requester can issue four 1 KB read requests or a single 4 KB read request. The 4 KB request results in higher throughput than the four 1 KB reads. The `Maximum Read Request Size` value in the `Device Control` register, bits [14:12], specifies the read request size.

Outstanding Read Requests

A final factor that can affect the throughput is the number of outstanding read requests. If the requester sends multiple read requests to improve throughput, the number of available header tags limits the number of outstanding read requests. To achieve higher performance, the Intel Arria 10 and Intel Cyclone 10 GX read DMA can use up to 16 header tags. The Intel Stratix 10 read DMA can use up to 32 header tags.

3.2.1. Understanding Throughput Measurement

To measure throughput, the software driver takes two timestamps. Software takes the first timestamp shortly after you type the `./run` command. Software then takes the second timestamp after the DMA completes and returns the required completion status, EPLAST. If read DMA, write DMA and simultaneous read and write DMAs are all enabled, the driver takes six timestamps to make the three measurements.

4. Document Revision History for AN 881: PCI Express Gen3 x16 Avalon-MM DMA with External and HBM2 Memories Reference Design

Document Version	Intel Quartus Prime Version	Changes
2019.07.19	19.1	Updated the Application Note to cover a single reference design for a PCI Express Gen3 x16 Avalon-MM DMA with external and HBM2 memories.
2018.12.27	18.1	Fixed typo in Figure 1.
2018.12.10	18.1	Initial release.