2015 The 5th International Workshop on Computer Science and Engineering
doi: 10.18178/wcse.2015.04.112

Design and Implementation of DMA Transfer Through PCI Express Gen3 Interface Based on FPGA

Yufei Zhang[1,a], Hengzhu Liu[1,b], Xiaodong Yi[1,c], Xuan Zhu[1,d]

[1]Department of Computer, National University of Defense Technology, ChangSha, 410073,China

[a] yueyizhuying@gmail.com, [b] hengzhu_liu@263.net, [c]xdong_yi@163.com, [d]zx00348711@163.com

Keywords: FPGA, PCI Express 3.0, DMA Read, DMA Write.

Abstract. The future requirements for bandwidth and scalability of big data has gone beyond the performance of the second generation I/O interconnect. Peripheral component interconnect express (PCI Express)3.0 is the latest standard as one of the third generation I/O interconnect for expansion cards. The bit rate of this fully serial interface has reached 8 giga transfers per second(Gbps), which could meet the demand of future computing and communication platforms. In this paper, we designed and implemented the directly memory access (DMA) transfer of PCI Express Gen3 interface based on the Xilinx Virtex-7 FPGA Gen3 Integrated Block for PCI Express core. The proposed design has debugged and verified on the Xilinx Virtex-7 FPGA VC709 Connectivity Kit and could carry an average rate of 3.04GB/s, this design provides up to 300% improvement over the similar design based on PCI Express Gen2 interface.

Introduction

With the development of cloud computing, internet of things and social network technology, the amount of data is growing and accumulating at a dramatically high speed. For example, Google accesses 400PB data per month through large-scale cluster and Map Reduce software. The rapid development of Big Data calls for a high speed of data transferring. In order to satisfy the demands of transferring rate, PCI Express is constantly developing. PCI Express 3.0 carries a data transferring rate of 8Gbps per lane and upgrades the coding scheme to 128b/130b[1]. It grows 3Gbps on date transferring rate and reduces the bandwidth overhead from 20% to 1.54% compared with PCI Express 2.0[2]. Since 2013, Xilinx, Altera, PLDA, NVIDA and other companies have released chips which in compliant with PCI Express 3.0 Base Specification, PCI Express 3.0 has been widely used in many fields[3], but currently these applications are not open source[4].

In this paper, we designed a PCI Express Gen3 interface through which the data could be transferred between the DDR3 in the computer and the SODIMM on the Xilinx VC709 development board. To improve the data transferring rata, we further designed a DMA controller. Overall, the major contributions of this paper are:

- We analyzed the upgrades of the Xilinx Virtex-7 FPGA Gen3 Integrated Block for PCI Express, and further designed a new PCI Express Gen3 interface based on this core.
- We designed a DMA controller based on the PCI Express Gen3 interface. The test results showed that this interface could carry a rate of 2.89GB/s of DMA read and 3.18GB/s of DMA write.

The remainder of this paper is organized as follows: In Section 2, we summarize the improvement of the Xilinx FPGA PCI Express core. Section 3 describes the design and implementation of PCI Express Gen3 DMA, followed by evaluation in Section 4, related work in Section 5, and conclusions in Section 6.

The improvement of Xilinx PCI Express Gen3 IP core

In this section, we made comparison between the Xilinx PCI Express Gen3 and Gen2 IP Cores. Through the analysis, we found that the performance advantages of theVirtex-7 PCI Express Gen3 Integrated Block originate from improvements in client interface, transaction layer packets(TLP) descriptor format and data alignment mode. Such a perspective provided us with a clear guidance of the design of the PCI Express Gen3 interface.

Client interface. TheVirtex-7 PCI Express Gen3 Integrated Block consists of four AXI4-Stream Interfaces o receive and transfer transactions while the Virtex-6 PCI Express Gen2 Integrated Block only have two AXI4-Stream Interfaces, thus the further one could simplify the transactions. Figure 1 shows the two different client-side interfaces.

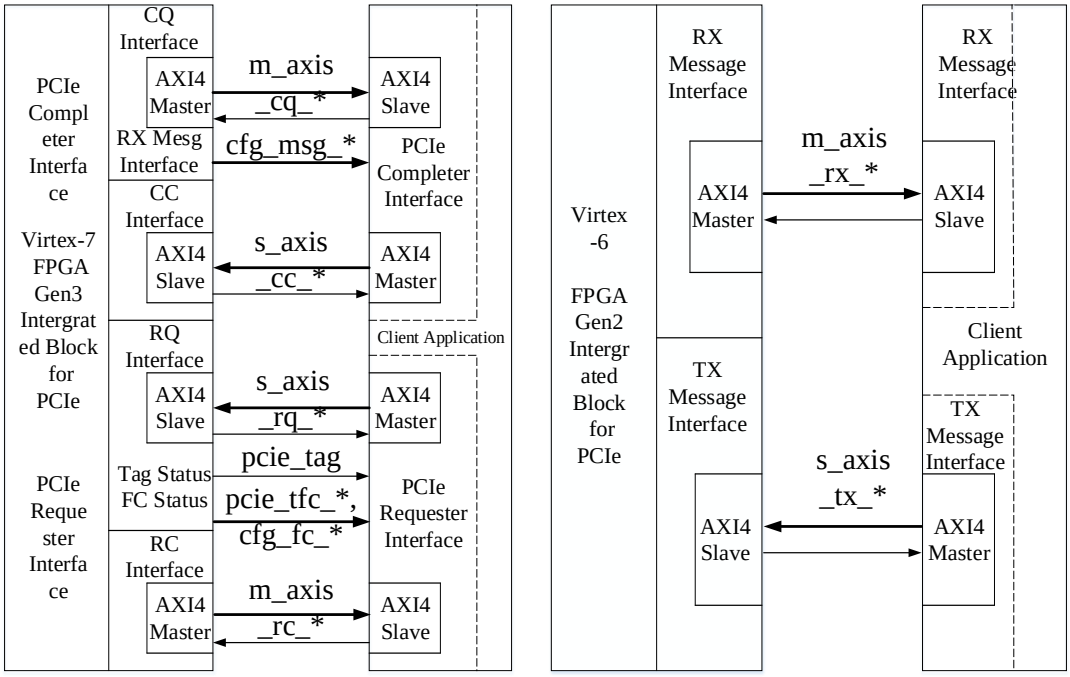

Fig 1: Block Diagram of Virtex-7 FPGA Gen3 Integrated Block Client Interfaces[5]and Virtex-6 FPGA Gen2 Integrated Block Client Interfaces[6]

- Completer reQuest (CQ) Interface: The interface delivers all of the request packets from the link to user application.
- Completer Completion (CC) Interface: The interface transmits the response packets generated by the user application. It can accept and send packets at the same time.
- Requester reQuest (RQ) Interface: The interface transmits the request packets generated by the user application.
- Requester Completion (RC) Interface: The interface delivers the response packets from the link.

The transactions on the Requester Interface and Completer Interface are similar, each of them has two separate interfaces, one for the different direction. Each of them is compliant with the AXI4-Stream protocol, and the width can be configured as 256,128, or 64 bits. The only difference is that the roles of the user application and the core are reversed.

TLP descriptor format. The Virtex-7 PCI Express Gen3 Integrated Block transmits and receives data as an independent packet. Every packet begin with a descriptor and can be followed with payload. The descriptor is sent in the first 4 double word (DW) and always 4DW long. During the first beat of the 256-bit interface, the descriptor is transferred. The format of memory request types descriptor is illustrated in Figure 2. The format of Figure 3 applies in the Virtex-6 PCI Express Gen3 Integrated Block interfaces.

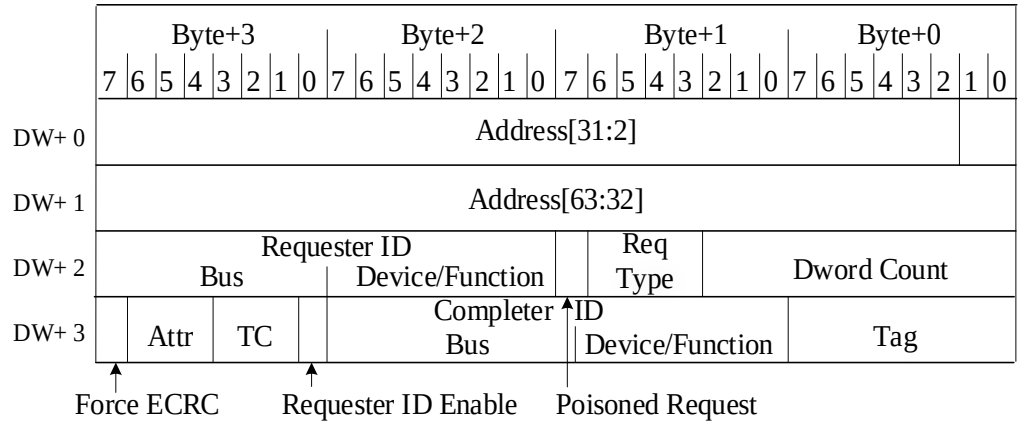

Fig 2: PCI Express Base Specification 3.0 Byte Order[1]

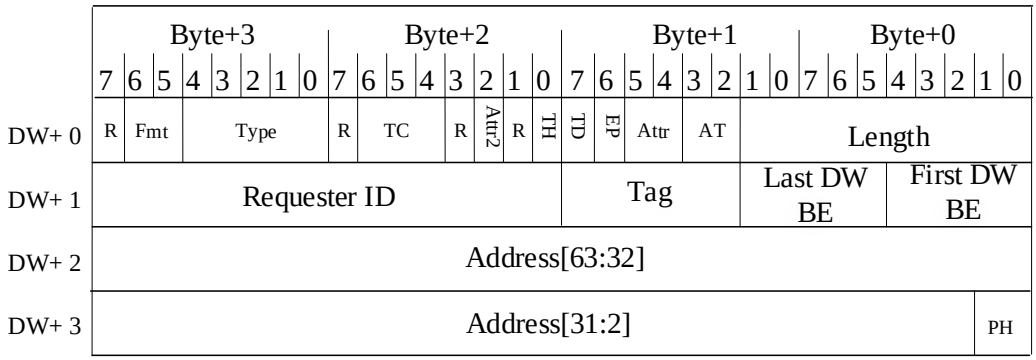

Fig 3: PCI Express Base Specification 2.0 Byte Order[2]

As the Virtex-7 PCI Express Gen3 Integrated Block consists of four interfaces based on the AXI4-Stream protocol, the completion packets from Root Complex (RC) no longer distinguished as MWr, Cpl or CplD, all of CplD packets can be delivered through the Requester Completion (RC) Interface specially. Then the Fmt byte defined in the PCI Express specification 2.0 can be removed. Table 1 shows the Gen3 and Gen2 transaction types.

Table 1: Gen3 Transaction Types[1]

IP cores	TLP Type	Request Type (binary)	Description
		Req Type[78:75]	
Gen3 IP Core	MRd	0000	Memory Read Request
	MWr	0001	Memory Write Request
	IORd	0010	I/O Read Request
	IOWr	0011	I/O Write Request

Table 2: Gen2transaction Types[1]

IP cores	TLP Type	Request Type (binary)		Description
		Fmt[2:0]	Type[4:0]	
Gen2 IP Core	MRd	000 001	0 0000	Memory Read Request
	MWr	010 011	0 0000	Memory Write Request
	IORd	000	0 0010	I/O Read Request
	IOWr	010	0 0010	I/O Write Request
	Cpl	000	0 1010	Completion without Data – Used for I/O and Configuration Write Completions with any Completion Status.
	CplD	010	0 1010	Completion with Data – Used for Memory, I/O, and Configuration Read Completions. Also used for Atomic Op Completions.

Data alignment mode. The TLPs transferred on the client interface consist of two parts: the descriptor and the payload data followed behind the descriptor (when it has one). When a TLP's payload is present, there are two data alignment mode to the data path:

• Dword-aligned Mode: The payload bytes are immediately follow behind the descriptor in the adjacent Dword position. For data transferred from the Completer Completion (CC) interface, the valid data is presented in lane N:

$$N = (12 + A \ mod \ 4) \ mod \ w \ [5]$$

where A is the starting address of the payload data being transferred in byte level, and the w is the interface width in byte.

• Address-aligned Mode: The payload bytes always starts in the next beat after the descriptor has transferred. For data transferred from the Completer Completion (CC) interface, the valid data is presented in laneN:

$$N = A \ mod \ w \ [5]$$

whereAis the memory address specified in the descriptor, w is the interface width in byte. If there are any gap between the descriptor and the first byte of the payload, it will be filled with null bytes.

From the above discussion, we demonstrates that theVirtex-7 PCI Express Gen3 IP core is more powerful and make the transfer much simpler.

The design and implementation of PCI Express Gen3 DMA

Our design is based on the Xilinx VC709 Connectivity Kit, and utilize the PCI Express Endpoint IP core in the xc7vx690t chip. In this design, the data could successfully transfer between the DDR3 in computer and the SO-DIMM on the VC709 development board. The details of this design as follows:

The design of DMA controller. The Virtex-7 PCI Express Gen3 Integrated Block supplies four client interfaces to users, which could meet user's needs better for designing a transaction layer. The DMAcontroller is the key of the transaction layer design.

The DMA controller consists of 6 parts: TX Engine、RX Engine、Tag Generator、Data Presenter、DMA Control/Status Regs and DMA DDR3 Interface, as shown in Figure 4.

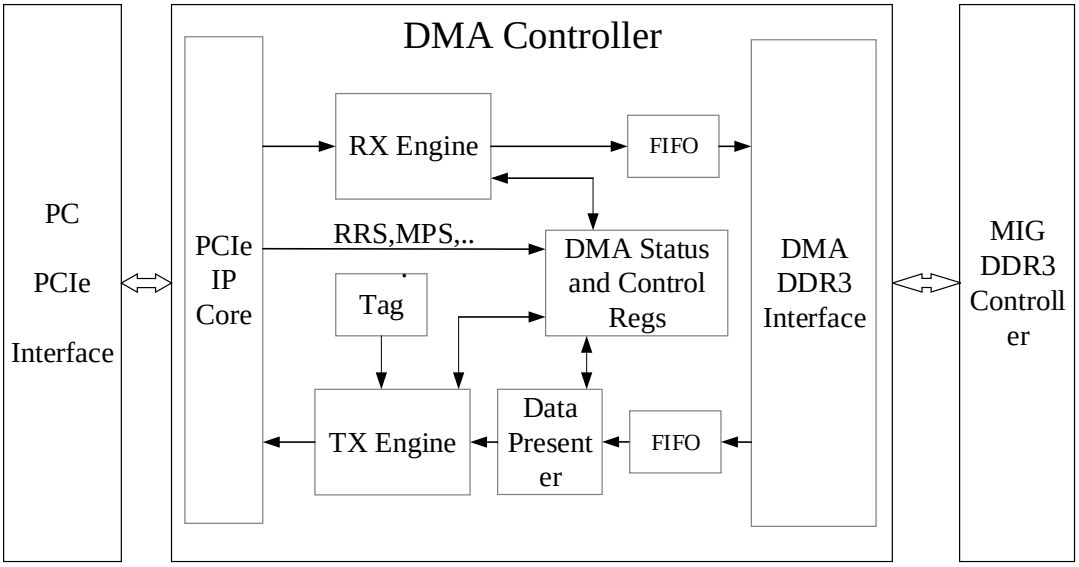

Fig 4： PCI Express Application Diagram

• RX Engine: The RX Engine receives and decodes the packets from the integrated block and routes the payload to the corresponding memory. Figure 5 shows the RX Engine State Machine.

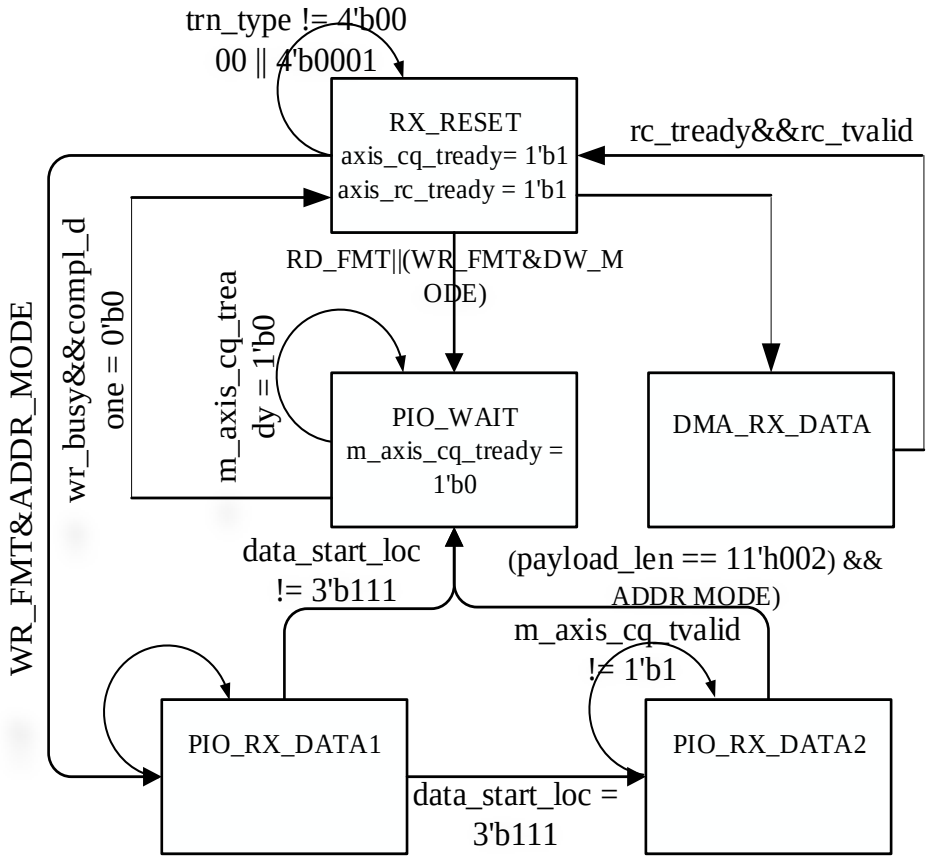

Fig 5：RX Engine State Machine

• TX Engine: The TX Engine generates and transmits the packets through a simple stare machine to the integrated block in compliant with AXI4-Stream protocol. Figure 6 shows the RX Engine State Machine.

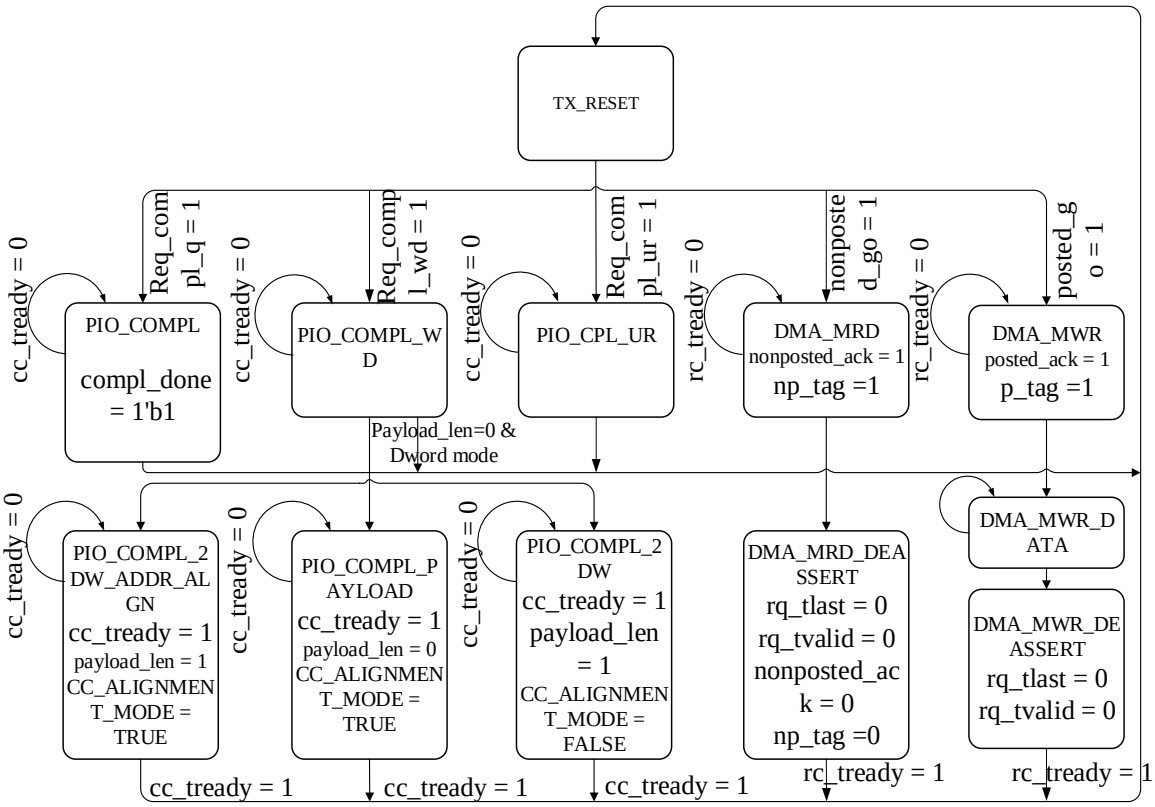

Fig 6：TX Engine State Machine

•Tag Generator: This block creates unique tags for each request packet. The tags are generated by one 8-bit counter. It is used to identify each outstanding packet in the link and the value mustn't confliction with the others[7].

•Data Presenter: This block provides 256-bit data from the DDR3 interface to the TX Engine block and utilize some handshaking signals to communicate with them.

•DMA Control/Status Regs: The DMA Control/Status Regsis mapped to the PCI memory space as the Base Address Register (BAR) 0, these regs define the DMA operations. All of these regs are 32-bit read/write access. It is the main block that the host communicates and control for DMA operations.

•DMA DDR3 Interface: The DMA DDR3 Interface provides MIG DDR3 address and other control signals, and make clock domain change between the MIG DDR3 clock domain and the DMA Controller domain.

The DMA read/write operation. The host executes a DMA read or write operation by initializing the DMA Control/Status Regs. The host sets the DMA read/write data source address, destination address, the transfer size and the start bit, then finish a complete DMA operation. The regs description are contained in Table 3.

Table3: DMA Control/Status Register Files Definition

Reg Number	Offset	Register Name	Types
1	0x00	Read DMA operation Start	DMA Read Operation Regs
2	0x04	Host Memory Source Upper	
3	0x08	Host Memory Source Lower	
4	0x0C	DDR3 Address Destination	
5	0x10	DMA Read Transfer Size	
6	0x14	Read DMA operation Done	
7	0x18	Write DMA operation Start	DMA Write Operation Regs
8	0x1C	Host Memory Destination Upper	
9	0x20	Host Memory Destination Lower	
10	0x24	DDR3 Address Source	
11	0x28	DMA Write Transfer Size	
12	0x2C	Write DMA operation Done	

The DMA read/write operation execute as follows: (1)Initialize the DMA read/write data source register; (2) Initialize the DMA read/write data destination register; (3) Initialize the DMA read/write transfer size; (4) Set the DMA read/write start bit; (5) Reset the DMA read/write start bit and set the DMA read/write done bit. The process is shown in Figure 7.

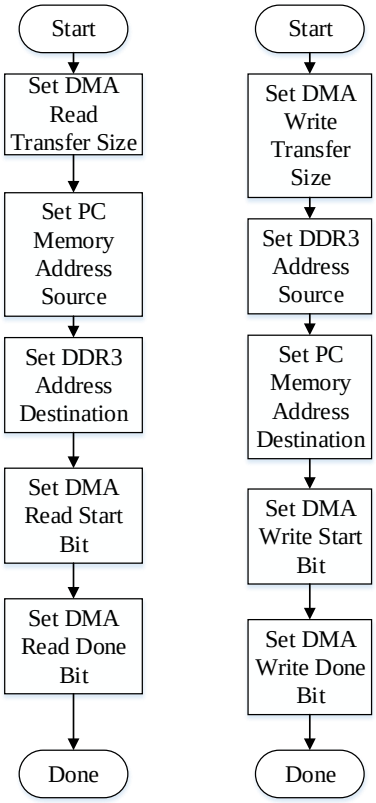

Fig 7: The DMA Read/Write Operation

Evaluation

The VIVADO Design Suit supplies a Graphical User Interface (GUI), in which the Virtex-7 PCI Express Gen3 Integrated Block can be configured. In this design, the Integrated Block is configured as shown in Table 4.

Table 4: The Configuration of PCI Express Gen3 Integrated Block

Configuration options	Parameters
Device/Port Type	PCI Express Endpoint device
Lane Width	x8
Maximum Link Speed	8.0GT/s
AXI-ST Interface Width	256 bit
AXI-ST Interface Frequency	250 MHz
AXI-ST Alignment Mode	Address Aligned
Reference Clock Frequency	250 MHz
PF0 Max Payload Size	256 Bytes
Bar0	8 KB(32bit)

Our design is based on the Xilinx VC709 Connectivity Kit, implemented and verified in VIVADO 2014.1 Design Suit, and tested under the Windows7 64-bit OS in DELL Vostro 270 host. The testing result is shown in Table 5.

Table 5: The Test Result of Bandwidth

Test Type		Speed
READ	RC Read Endpoint (Single DW)	1.5MBps
	Endpoint Read RC (DMA)	2.89GBps
WRITE	RC Write Endpoint (Single DW)	3.0MBps
	Endpoint Write RC (DMA)	3.18GBps

Related work

This section summarized the prior work on the design and implementation of DMA transfer through PCI Express interface based on FPGA. Recently there has been a lot of research for designing the DMA controller through PCI Express Gen2 interface. [8], implemented a DMA reading and writing module on the Xilinx Virtex-5 based on the PCI Express 2.0 protocol. In[9], the authors have proposed a design of DMA transmission with PCI Express 2.0 interface based on the Altera Cyclonc IV GX.

Our work differs from all these prior works in that, we designed a new PCI Express Gen3 interface based on the PCI Express 3.0 protocol and designed a DMA controller based on the Xilinx Virtex-7 FPGA Gen3 Integrated Block for PCI Express.

Conclusions

In this paper, we designed a new PCI Express Gen3 interface based on the Virtex-7 PCI Express Gen3 Integrated Block and further designed a DMA Controller based on the PCI Express Gen3 interface. The test results show that this interface could carry an average rate of 3.04GB/s, this design provides up to 300% improvement over the similar design based on PCI Express Gen2 interface.

Acknowledgements

This work is supported in part by Open Research Fund of State Key Laboratory of High Performance Computing under Grant No.201302-01 and No.201402-01 andthe SRFDP under Grant No.20114307130003 and the National Natural Science Foundation of China under Grant No.61176030.

Reference

[1] PCI-SIG. PCI Express Base Specification Revision 3.0 [Z]. https://www.pcisig.com/specifications/pciexpress/base3/.

[2] PCI-SIG. PCI Express Base Specification Revision 2.0 [Z]. https://www.pcisig.com/specifications/pciexpress/base2/.

[3] PCI-SIG. PCI Express 3.0 Integrators List[Z].https://www.pcisig.com/developers/compliance_program/integrators list/pcie_3.0/.

[4] Virtex-7 FPGA XT Connectivity Targeted Reference Design for the VC709 Board. http://www.xilinx.com/support/documentation/boards_and_kits/vc709/2013_2/ug962-v7-vc709-xt-connectivity-trd-ug.pdf

[5] Virtex-7 FPGA Gen3 Integrated Block for PCI Express v3.0 LogiCORE IP Product Guide.http://www.xilinx.com/support/documentation/ip_documentation/pcie3_7x/v3_0/pg023_v7_pcie_gen3.pdf

[6] Virtex-6 FPGA Integrated Block for PCI Express User Guide. http://www.xilinx.com/support/documentation/ip_documentation/v6_pcie/v2_5/ug671_V6_IntBlock_PCIe. pdf

[7] MindShare, Inc, Ravi Budruk, Don Anderson, Tom Shanley. PCI Express System Architecture[M]. Addison Wesley. September 04, 2003.

[8] Jinghua Wang, Shanqing Hu. The Implementation of DMA Reading and Writing Module on FPGA Based on PCI Express Protocol[J]. Microcomputer Information,2010,26(10):7-9.

[9] Muguo Li, Ying Huang. Design of DMA Transmission with PCIe Bus Interface Based on FPGA[J]. Computer Measurement & Control,2013.21(1):233-235.