

Design & Implementation of PCI Express BUS Physical layer using VHDL

Ankita R. Tembhare

Research Scholar
JDCEM Nagpur University,
Maharashtra, India
atembhare3@gmail.com

Dr.Pramod B. Patil

Principal
JDCEM, Nagpur University,
Maharashtra, India
ppamt07@yahoo.com

Abstract:-This paper presents the proposal of the implementation of the Physical Link Layer of PCI-Express, as is defined in PCI Express1.0.The architecture presented here contains the transmission and receiver modules which ensure the reliably conveying of the Transaction Layer Packet (TLP) and Data link Layer Packet(DLLP) between two components using the PCI-Express protocol. This paper explains how the implementation makes the reliably conveying ,through the addition of a start and end bits to each data coming in from the Transaction and Data link Layer in the transmit side, and how the packets are processed in receiver side. The whole architecture will be implemented on Altera or similar FPGAs to indicate that this architecture is a feasible approach. PCI Express bus architecture will be implemented on VHDL platform

Keywords- PCI Express, TLP, DLLP, FPGA, Xilinx, vertex.

1. Introduction:

PCI Express is the third generation high performance I/O bus used to interconnect peripheral devices in applications such as computing and communication platform. PCI Express is a high speed serial computer bus standard designed to replace the older PCI, PCI-X bus standard.

Most state of the art PCI have new bus called PCI Express. It is a 3rd generation high performance I/O Bus used to interconnect peripheral devices in applications such as mobile,desktop, workstations ,server, a serial, point to point type interconnect for communication platforms. PCI Express technology implements a serial interconnect between two devices result in fewer pins for device package, which reduces the PCI Express chip ,board design cost and design complexity. PCI Express implement switch based technology to interconnect a large number of devices. Communication over the serial interconnect is accomplished using a packet based communication protocol. Physical layer link should be configured varying from 1-32 lanes ,with each lane carrying a maximum data rate of 2.5GB/s.

Since the first PC, launched in 1981, the computer has had expansion slots where you can install additional cards to add capabilities not available on the motherboard of the computer. Currently, the most common type of expansion slot available is called PCI Express PCI stands for Peripheral Component Interconnect. PCI Express has been launched by Intel in 2004 to replace the PCI extension bus. This bus is used as a communication lane to transmit signals and data from your computer system to peripheral devices attached to your computer. Hence, PCI Express, also known as 3GIO, is a computer expansion card interface, will transmit data along four point-to-point serial data lanes.

PCI Express, alike Gigabit Ethernet, SATA, and Serial-Attached SCSI, will make use of a high-speed serial link technology to be able to cope with advanced processors and I/O technology.

PCI Express is structured in layers that will be used to run read and write requests that are transmitted by the transaction layer to the I/O devices. This transaction is done through a packet-based and splitting process. The bus bandwidth is different from the normal PCI one as it is not shared but is supplied to each device PCI Architecture has become outdated. PCI Express slots are found on many motherboards, letting computer users install components into them.

PCI Express bus runs in serial interface, which allows it to reach a bandwidth that is much higher than that PCI bus.PCI is a bus, whereas PCI Express is a point to point connection, i.e , it connect only two devices; no other device can share this connection. Just to clarify, on a motherboard using standard PCI slots, all PCI device are connected to the PCI bus and share the same data path ,so a bottleneck (i.e performance decrease because more than one device wants to transmit data at the same time) may occur.

On a motherboard with PCI Express slot, each PCI Express slot is connected to the motherboard chipset using a dedicated lane, not sharing this lane(data path) with other PCI Express slots. Also devices integrated on the motherboard, such as network, SATA and USB controllers, are usually connected to the motherboard chipset using dedicated PCI Express connections.

PCI and all other kinds of expansion slots use parallel communications, while PCI Express is based on high speed serial communications.PCI Express is based on individual lanes, which can be grouped to create higher bandwidth connections. The “x” that follows the description of a PCI Express connection refers to the number of lanes that connection is using.

The PCI Express connection represents an extraordinary advance in the way peripheral devices communicate with the computer. It differs from the PCI bus in many aspects, but the most important one is the way data is transferred.

The PCI Express connection is another example of the trend of migrating data transfer from parallel communication to serial communication. Other common interfaces that use the serial communication include the USB, the Ethernet (networking) and the SATA and SAS (storage).

PCI Express protocol follows a layered structure similar to the OSI Model and contains the following link layers: software layer, transaction layer, data link layer and physical layer.

2. Review of PCI Express Bus

PCIe is a third generation high speed I/O bus used to interconnect peripheral devices in applications such as computing and communication platforms.

PCIe employs point-to-point interconnects for communication between two devices as against its predecessor buses that used multi-drop parallel interconnect. A point-to-point interconnect implies limited electrical load on the link, allowing higher frequencies to be used for communication. Currently, with the PCIe Gen2 specifications, the link speed is 5 Gbps as against the Gen1 with 2.5 Gbps speed. Because of the serial interconnection, the board design cost and complexity has reduced considerably.

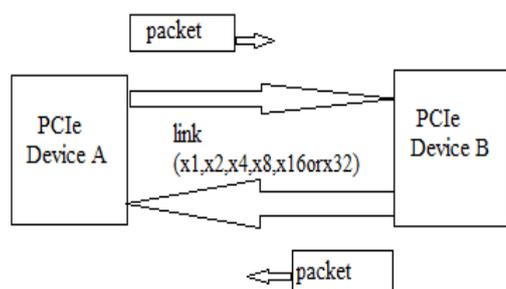


Figure 1: Pci Express link

A PCIe interconnect that connects two devices together is referred to as a link. A link consists of either x1, x2, x4, x8, x16 or x32 signal pairs in each direction. These signals are referred to as Lanes. Figure 1 shows how data is transmitted from a device on one set of signals and received on another set of signals. PCIe, unlike previous PC expansion standards, is structured around point-to-point serial links, a pair of which (one in each direction) make up a lane rather than a shared parallel bus. These lanes are routed by a hub on the main board acting as a switch. This dynamic point-to-point behavior allows more than one pair of devices to communicate with each other at the same time. In contrast, older PC interfaces had all devices permanently wired to the same bus;

therefore, only one device could send information at a time.

3. Methodology:

Understanding of the PCI Express 1.0 Physical Layer Specification. To design an Architecture from the specification. Behavioral /RTL modeling of Design blocks. Design of stimulus modules to test the functionality of Design blocks. Synthesize design to extract Gate level net list. Use the net list to target (implementation) FPGA.

a: Design Implemented

In this section we describe our implementation of the PCI Express Physical layer. This deals with the design of physical layer transmit and receive protocols which connect to the link on one side and connect to the data link layer on the other side.

The proposed Physical layer architecture has two modules, the transmitter and the receiver.

1: Transmitter

It essentially processes a packet arriving from the data link layer and then converts it into a serial bit stream. The bit stream is clocked out at 2.5 GB/s per lane onto the link. It frames the TLPs or DLLPs with start and end characters with the help of a control block.

With the aid of a multiplexer, the Physical Layer frames the TLPs or DLLPs with start and end characters. These characters are framing symbols which the receiver device uses to detect the start and end of a packet.

A Scrambler uses an algorithm to pseudo-randomly scramble each byte of the packet. Scrambling eliminates repetitive patterns in the bit stream. Repetitive patterns result in large amounts of energy concentrated in discrete frequencies, which leads to significant EMI noise generation. Scrambling spreads energy over a frequency range, hence minimizing average EMI noise generated.

The 8b/10b encoder encodes scrambled characters into 10-bit symbols. These 10-bit symbols are converted to the serial bit stream by using a parallel-to-serial converter.

2: Receiver

At the receiver, the serial bit stream is given to a serial-to-parallel converter. The 10-bit symbols are converted back to 8-bit characters by the 8b/10b Decoder.

The De-Scrambler reproduces the de-scrambled packet stream from the incoming scrambled packet stream. The De-Scrambler implements the inverse of the algorithm implemented in the transmitter Scrambler.

Figure 2 shows the block representation of the PCI Express 1.0 physical layer structure. It represents the data path flow inside the layer. It has a Transmitter and a Receiver part.

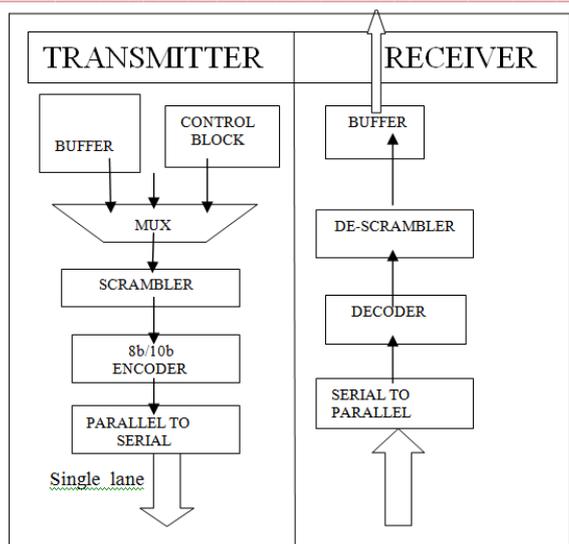


Figure 2: PCI Express Physical Layer

B: Software Implementation

The above figure shows the PCI Express Bus Physical Layer x1, In our work Input data packets coming from the higher layers are stored in the buffer of PCI Express physical layer Transmitter, than this data packets are framed with start and end byte, and after scrambling and encoding the data packets are transfer to the PCI Express Physical layer Receiver serially. The Start and End framing bytes are not scrambled.

The 2:1 Mux at the transmitter is used to select between the data and start and end framing characters. When control signal of mux is high framing character is the output of mux. when it is low the data is at the output of the mux.

Simulating the written VHDL code in ISE simulator and verifying the waveforms generated by the simulator. If the required simulated output is not achieved, then the VHDL code is checked and necessary corrections are made. The required waveforms are noted down as a reference to the synthesis stage or at the final stage. The waveforms are presented subsequently

Device utilization summary:

Number of Slices:	265 out of 1920	13%
Number of Slice Flip Flops:	312 out of 3840	8%
Number of 4 input LUTs:	396 out of 3840	10%
Number of IOs:	21	
Number of bonded IOBs:	21 out of 173	12%
Number of GCLKs:	1 out of 8	12%

Thus I had implemented the PCI Express Bus Physical layer XILINX ISE 9.1. The RTL VIEW and Simulation waveforms are shown below

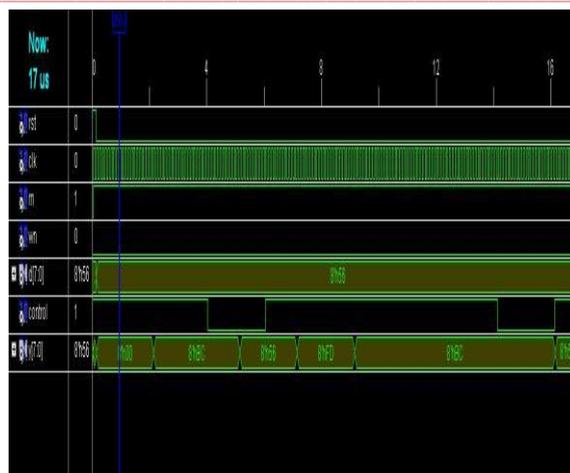


Figure 3: Simulation Result

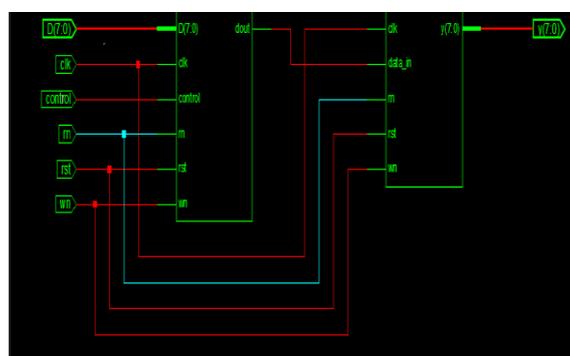


Figure 4: RTL View Of Complete System

C: Hardware Implementation

An altera Cyclone II FPGA device is used for this implementation .The system is tested for transmission of data through PCI Express bus Physical Layer. The toggle switches are used as an input to the system. An internal clock pulse of 27MHz is used which is designated as pin D13 in DE2 board. The DIP Switches SW0 to SW7 are used as input to the system. Red LED's from LEDR(0) to LEDR(7) are used for representing the output of the system.SW13 switch is used as an active high reset for the syetem.SW16 and SW15 are used for read and write signal to the system.SW17 is used as control pin, when it is high it indicate the star and end frame and when it is low it indicate the data is transmitted.

All the VHDL module files are integrated and a top file is selected to run the whole implementation. Implementation compresses of different stages like translate, mapping. A programming file is generated with extension name .SOF which creates a bit format file, which is dumped on the altera Cyclone II FPGA using Quartus II (software) through a USB Blaster. On the hardware, if the design works according requirement, then design is corrected, if any error occurs then reconsider the design.

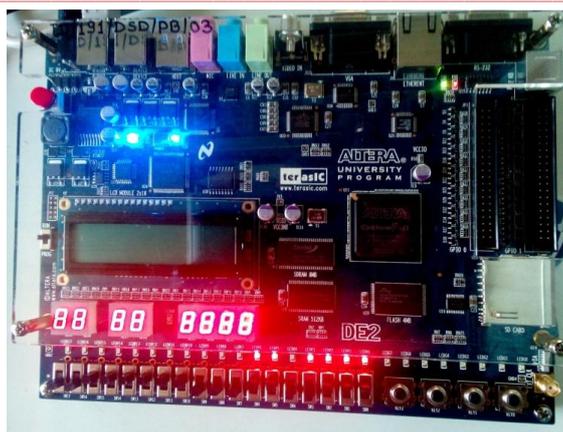


Figure 5: Hardware Implementation Result

4: CONCLUSION

Thus I had implemented the design of PCI Express bus Physical layer architecture using VHDL language and to make an analytical evaluation with the help of Xilinx 9.1 ISE tool. I had implemented Physical layer Transmitter and Receiver communicating via FPGA. The synthesis part has been performed on Xilinx ISim Simulator and the Altera DE2 Board Cyclon II family FPGA. The results shown below depicts that Peer to peer interconnection which transmits data in a serial manner between the physical layer of PCI Express bus.

6. References:

[1] PCI Express Base Specification, <http://www.pcisig.com>

- [2] R. Bittner, "Speedy bus mastering PCI Express", in FPL. IEEE,2012,pp.523-526
- [3] M.Jacobsen, Y.Freund,R. Kastner, "RIFFA Reusable Integration Framework for FPGA Accelerators", in FCCM IEEE Computer Society,2012,pp.216-219
- [4] Yingjie cao & yangxin Zhu,Xu Wang & Jiang,Meikang Qui "An FPGA based PCI Express Root Complex Architecture for Standalone SOPC's" IEEE 2013
- [5] PCI Express Base Specification, <http://www.pcisig.com>
- [6] C. Gao, S. Lu, "Novel FPGA based Haar classifier face detection algorithm acceleration", in FPL. IEEE, 2008, pp. 373-378
- [7] PCI Express System Architecture (Book) PCI Express System Architecture (Book) by Ravi Budruk ,Don Anderson ,Tom Shanley
- [8] Sarun O.S. Nambiar, Yogindra Abhyankar, Sajish Chandrababu "Migrating FPGA based PCI Express Gen1 design to Gen2" ICCCT'10
- [9] Kadric, N. Manjikian, Z. Zilic, "An FPGA implementation for high-speed optical link with a PCIe interface", in SOCC. IEEE, 2012, pp. 83-87
- [10] Endpoint for PCIe user guide (SofIP)
- [11] Shaghayegh Abiri, Sara Sahebdeh," A method for implementing of the DC Balanced 8B\10B coding used in super spread USB 2010 International conference on Integrated Intelligent Computing.
- [12] Hu li Yuan'an liu, Dongming Yaun, Hefei Hu "A Wrapper of PCI Express with FIFO Interfaces based on FPGA" 2012 International conference on Industrial control and Electronics Engineering.
- [13] Liu Qihao Weig Huihui and Zhang Fing, Zhoo Jinazhong "An Efficient Physical coding Sublayer for PCI Express in 65 nm Cmos" 2012 IEEE International Symposium on Intelligent Signal Processing and Communication System