International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Email:editorijoer@gmail.com http://www.ijoer.in

Vol.4., Issue.3., 2016
(May-June)

RESEARCH ARTICLE

ISSN: 2321-7758

DESIGN AND IMPLEMENTATION OF A PCI EXPRESS PHYSICAL LAYER TRASMIT PROTOCAL

M.SIVAKUMAR[1],P.NAGAMALLAIAH[2],P G VARNA KUMAR REDDY[3]
[1,2,3]Academic assistant,
ECE Department, JNTUACEP, Pulivendula, Kadapa. A.P.

ABSTRACT
The project deals with the design of PCI Express physical layer transmit protocol, which connects to the link on one side and connects to the data link layer on the other side. It essentially processes packets arriving from the data link layer , and converts them into serial bit stream. The bit stream is clocked out at 2.5Gbits/sec per lane onto the link. The physical layer frames and deframes the TLPs and DLLPs with start and end characters. In Transmit logic the framed packet is sent to the byte stripping logic, which multiplexes the bytes of the packet onto the lanes and performing the CRC check on the payload data.

The Receive logic deframes the packet and perform descramble, 8b/10b decoding, coverts serial to parallel data. Scrambler uses an algorithm to pseudo-randomly scramble each byte of the packet. The 8b/10b encoder encodes scrambled characters into 10-bit symbols, which are converted to serial bit stream by the parallel to serial converter.

1. INTRODUCTION

PCI express is a 3rd generation high performance I/O bus used to interconnect peripheral devices in applications such as mobile, desktop, workstations, server, embedded computing and communication platforms. In PCI express, high speed links can currently transfer up to 2.5Gbps, and future implementations of PCI express will support as much as 10Gbps. PCI express technology implements a serial, point-to-point kind interconnect for communication between 2 devices. A serial interconnect between 2 devices results in fewer pins for device package, which reduces the PCI express chip, board design cost and design complexity.

PCI express implements switch-based technology to interconnect an oversized variety of devices. Communication over the serial interconnect is accomplished using a packet based communication protocol. Physical layer link could be configured varying from 1-32 lanes, with each lane carrying a most data rate of 2.5Gbits/sec PCI express protocol follows a layered structure similar to the OSI model and contains of the following four layers: software Layer, transaction Layer, link Layer, and Physical Layer.

Point-to-point interconnects: PCI express interconnect consists of either a x1, x2, x4, x8, x12, x16 or x32 point-to-point Link. A PCI express Link is that the physical association between just two devices. A Lane consists of signal pairs in every direction. an x1 Link consists of 1 Lane or 1 differential signal pair in every direction for a total of 4 signals. an x32 Link consists of 32 Lanes or 32 signal pairs for every direction for a total of 128 signals. Note that the Link only supports a symmetric number of Lanes in every direction, and does not support asymmetric topologies that would have more lanes from Device A than are sent by Device B in return. Throughout hardware initialization, the Link is automatically initialized for Link width and frequency of operation by the devices on opposite ends of the Link. Neither the OS nor computer code is firmware during Link level formatting.[1]

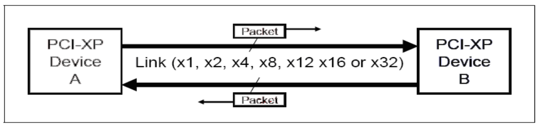

Figure 1.1:Link in terms of Lanes

2. PCI EXPRESS

2.1 PCI Express Topology:

The major components of the PCI express system shown in Figure 1.4 include a root complex, switches, and endpoint devices. A Root complex connects the cpu and memory subsystem to the PCI express fabric. it may support several PCI express ports, and this instance shows it supporting three ports. each port is connected to an end point device or else to a switch that then forms a sub hierarchy. the root complex generates transaction requests on behalf of the cpu. In response to cpu commands, it generates configuration, memory and IO requests as well as locked transaction requests on the PCI express fabric. the root complex transmits packets out of its ports and also receives packets into its ports which it then forwards to memory or the cpu. A multi-port Root advanced may optionally route packets from one port to another port (supporting peer-to-peer transactions) but is not required by the specification to do so legacy Endpoints might support IO transactions, and should support locked transaction semantics as a completer however not as a requester. Interrupt-capable bequest devices might support bequest style interrupt generation using message requests however also support MSI generation exploitation memory should write transactions. Bequest devices aren't needed to support 64-bit memory addressing capability.

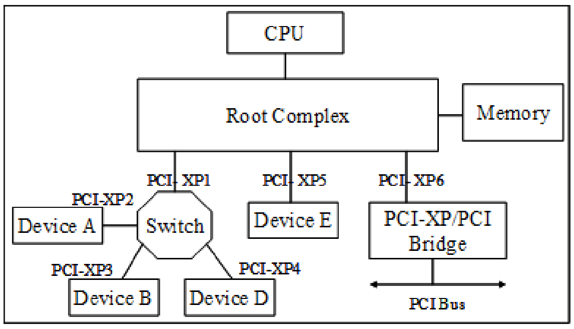

Figure 2.1: PCI Express Topology

2.2 PCI EXPRESS LAYERS:

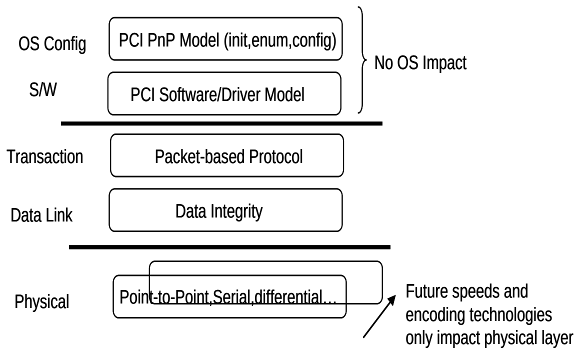
OS Config
S/W
PCI PnP Model (init,enum,config)
PCI Software/Driver Model
No OS Impact
Transaction
Packet-based Protocol
Data Link
Data Integrity
Physical
Point-to-Point,Serial,differential...
Future speeds and encoding technologies only impact physical layer

Figure 2.2: PCI Express Layered Architecture

The PCI express architecture is specified in layers, as shown in figure 2.2. Compatibility with the PCI addressing model load-store architecture with a flat address space is maintained to ensure that all existing applications and drivers operate unchanged. The PCI express configuration uses customary mechanisms defined within the PCI plug-and-play specification.

The software layers will generate read and write requests that are transported by the transaction layer to the I/O devices employing a packet-based, split-transaction protocol. The link layer adds sequence numbers and CRC to these packets to create a highly reliable data transfer mechanism. the essential physical layer consists of a dual simplex channel implemented as a transmit pair and a receive pair[1][2].

2.2.1 Physical Layer

The fundamental PCI express link consists of two low-voltage AC-coupled differential pairs of signals (a transmit pair and a receive pair). The physical link signal uses a de-emphasis scheme to scale back so symbol interference, to reduce rising information integrity. a information clock is embedded using the 8b/10b encoding scheme to achieve very high data rates. The initial signal frequency is 2.5Gb/s/direction and this is expected to increase with advances in silicon technology to 10 GB/s/direction (the practical maximum for signals in copper). The physical layer transports packets between the link layers of two PCI express agents. The bandwidth of a PCI express link may be linearly scaled by adding signal pairs to form multiple lanes

PCI links are 8b/10b encoded. This has the net effect of reducing usable PCIe lane bandwidth to 2 gigabits. PCIe uses the commonly used 10-bit control codes to communicate at this layer. Such communication includes link width negotiation, lane reversal, and packet delimiting. All PCIe packets are 32-bit aligned and the K code delimiters are included in that alignment restriction.

2.2.2 Data Link Layer

The primary role of a link layer is to confirm reliable delivery of the packet across the PCI express link(s). The link layer is responsible for information integrity and adds a sequence number and a CRC to the dealing layer packet as shown in Figure two.3 below. Most packets area unit initiated at the transaction layer. A credit-based, flow control protocol ensures that packets area unit transmitted only it's known that a buffer is available to receive this packet at the opposite end, that eliminates any packet retries and also the associated waste of bus bandwidth due to resource constraints. The link layer can automatically retry a packet that was signaled as corrupted. the information link layer adds data integrity features

2.2.3 Transaction Layer

The transaction layer receives read and write requests from the software layer and creates request packets for transmission to the link layer. All requests are implemented as split transactions and some of the request packets require a response packet. The transaction layer also receives response packets from the link layer and matches these with the original software requests. Each packet has a unique identifier that enables response packets to be directed to the correct originator. The packet format offers 32-bit memory addressing and extended 64-bit memory addressing. Packets also have attributes such as “no-snoop,” “relaxed ordering,” and “priority,” which may be used to route these packets optimally through the I/O subsystem.

The transaction layer provides four address spaces – three PCI address spaces (memory, I/O and configuration) and message space. PCI 1.4 introduced an alternate method of propagating system interrupts called message signaled interrupt (MSI). Here a special-format memory-write transaction was used instead of a hard-wired sideband signal, as an optional capability in a PCI 1.4 system. The PCI Express specification reuses the MSI concept as a primary method for interrupt processing and uses a message space to accept all prior sideband signals, such as interrupts, power-management requests, and resets, as in-band messages. Other “special cycles” within the PCI 1.4 specification, such as interrupt acknowledge, are also implemented as in-band messages. You could think of PCI Express messages as “virtual wires” because their effect is to eliminate the wide array of sideband signals currently used in a platform implementation

3. DESIGNING OF PCI EXPRESS

3.1 Design Blocks

3.1.1 TLP FIFO, DLLP FIFO

FIFO’s are used commonly in electronic circuits for buffering and flow control. In hardware form a FIFO primarily consists of a set of read and writes pointers, storage and control logic. A synchronous FIFO is a FIFO where the clock is used for both reading and writing. Examples of FIFO status flags include: full, empty, almost full, or almost empty. In hardware FIFO is used for speed synchronization purposes. FIFO Empty: When difference of read pointer and write pointer equal to the length of the FIFO, then it triggers the FIFO Empty signal. FIFO Full: When read pointer and write pointer are equal, and then it triggers the FIFO Full signal.

we use two types of FIFO’S: TLP FIFO (32x8) and DLLP FIFO (6x8).

TLP FIFO

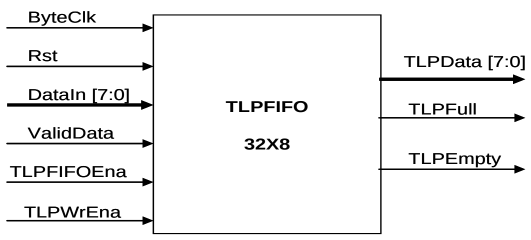

Figure 3.1(a) Top level Module of TLP FIFO

DLLP FIFO

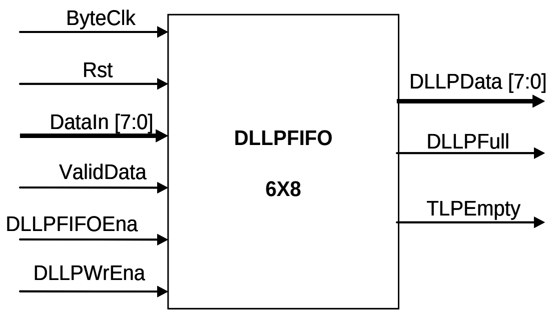

Figure 3.1(b) Top level Module of DLLP FIFO

Functional Description:

From The Figures 3.1(a) and 3.1(b) TLP's and DLLP's (packets) from the data link layer are clocked into a transmit buffer using TLP FIFO and DLLP FIFO. With the aid of multiplexer the physical layer frames the TLP's or DLLP's with start and end characters. These characters are framing symbols which the receiver device uses to detect start and end of packet. The transmit buffer uses a throttle signal to stop the flow of packets from the data link layer.

3.1.2 SCRAMBLER

Scrambler Algorithm:

The scrambler in figure 3.1.2 is implemented with a 16-bit Linear Feedback Shift Register (LFSR) that implements the polynomial:

$$G(x) = X^{16}+X^{5}+X^{4}+X^{3}+1$$

The LFSR is clocked at the bit transfer rate. The LFSR output is serially clocked into an 8-bit register that is XORed with the 8-bit characters to form the scrambled data.

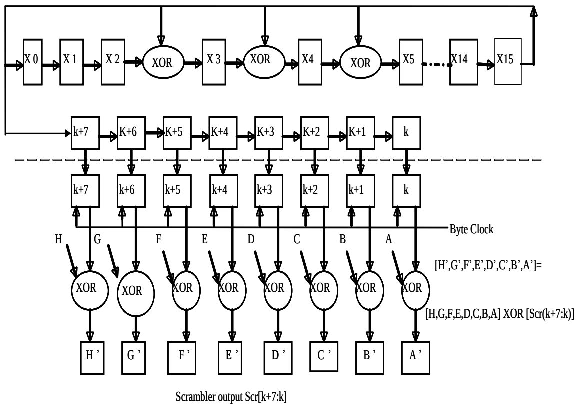
X 0
X 1
X 2
XOR
X 3
XOR
X4
XOR
X5
X14
X15
k+7
K+6
K+5
K+4
K+3
K+2
K+1
k
Byte Clock
H
G
F
E
D
C
B
A
[H',G',F',E',D',C',B',A']=
[H,G,F,E,D,C,B,A] XOR [Scr(k+7:k)]
H'
G'
F'
E'
D'
C'
B'
A'
Scrambler output Scr[k+7:k]

Figure 3.1.2 Scrambler Block diagram

Scrambler implementation rules:

- Scrambling is applied to 'D' characters associated with TLP and DLLP's, including the Logical Idle (00h) sequence. 'D' characters within the TS1 and TS2 Ordered-Set are not scrambled.
- 'K' characters and characters within Ordered Sets such as TS1, TS2, SKIP, FTS and Electrical Idle Ordered-Sets are not scrambled. These characters bypass the scrambler logic.
- Compliance Pattern related characters are not scrambled.
- When a COM character exits the Scrambler, (COM does not get scrambled) it initializes the LFSR. The initialized value of the 16-bit LFSR is FFFF h. Similarly on the receiver side, when a COM character enters the De-Scrambler, it is initialized.

By default, Scrambling is always enabled. Although the specification does allow the Scrambler to be disabled for test and debug purposes, it does not provide a standard software or configuration register-related method to disable the Scrambler.

3.1.3 STATE MACHINE CONTROLER (SMC)

Description: By the Figure 3.1.3 At start, When Rst is active state machine goes to IDLE state and it will reside in that state until 'TxEna' signal is activated. When this 'TxEna' signal asserted this goes to ByteStripStart State. This state is for sending first three bytes of the TLP or DLLP by appending respective start character depending upon 'TLPPackEna' or 'DLLPPackEna' signals. Control will be resided in this state for 9 clock pulses. For 10th Clock pulse Encoding state will be reached and for the next clock pulse SMC will generate control signals corresponding to retrieve remaining bytes of the Packet.

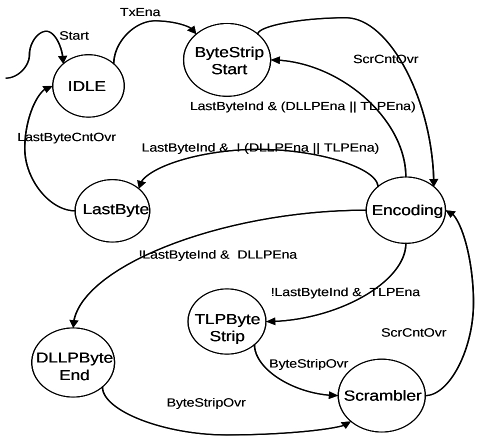

Figure
3.1.3: State Machine Diagram of Transmitter

Now, if it is 'TLPPackEna' asserted control goes to TLPByteStrip state otherwise it goes to DLLPByteEnd state. In TLPByteStripState it will be for 4 clock pulses and for 5 clock pulses in the Scrambler state and for Last clock pulse in the Encoding state. Like this, for sending any single byte on to single lane it will take 10 clock pulses. Finally, for sending 4 bytes of data on to 4 Lane Link we require 10 clock pulses.

In TLPByteStrip state, if 'TLPByteCntOvr' signal is asserted it will append END characters as the last byte of the packet. Similarly, in DLLPByteEnd state also END character will be appended and goes to Encoding state. In Encoding state control signals are issued to do Encoding and serialization. And immediately it has to come to other state. But to keep up this serialization signals until last bytes have transmitted we selected another state that is LastByte state. In this state, simply it counts 10 clock pulses to send the Last four 10-bit symbols and then goes to IDLE state. Until another packet comes control will be in that state only.

4 SIMULATION RESULTS

In the Figure 4.3 circle indicates initializing all input data. It means that 0ns to 1000ns all are input data. After 1000ns the output data will start in binary right shifting data.

In the Figure 4.4 circle indicates initializing all input data. It means that 0ns to 1000ns all are input data. After 1000ns the output data will start in binary right shifting data.

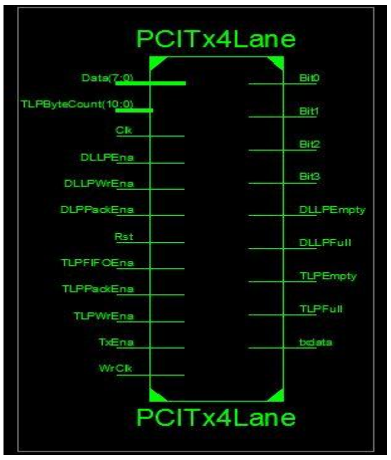

Fig 4.1 Top module of PCITx4Lane

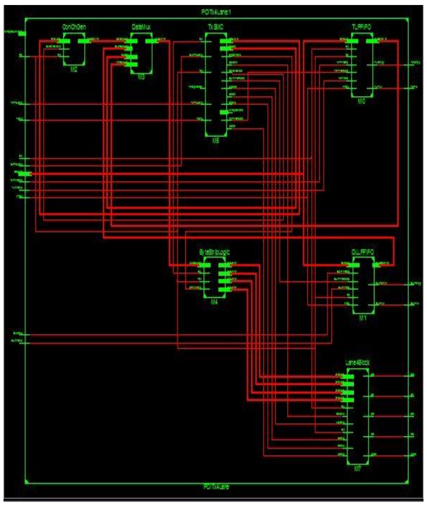

Fig 4.2 RTL Schematic of PCITx4Lane

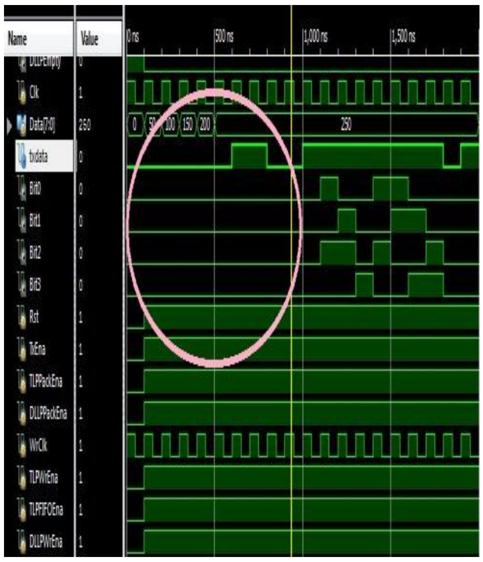

Figure 4.3: output simulation results by giving 50,100,150,200 input data

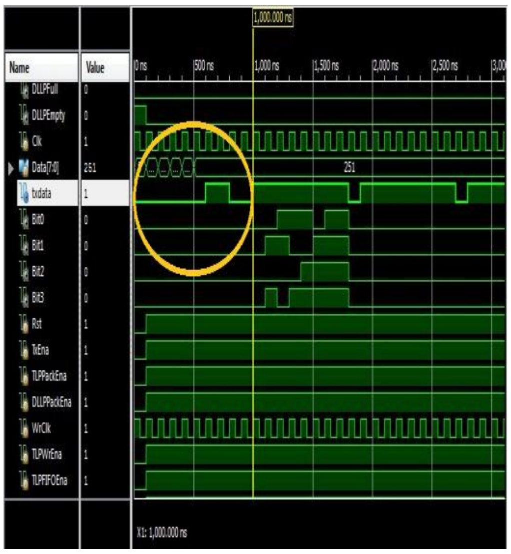

Figure 4.4: output simulation results by giving 220,230,240,250 input data

5 CONCLUSION

Various individual modules of PCI EXPRESS Transmitter and Receiver have been designed, verified functionally using VerilogHDL (ModelSim), synthesized by the XILINX synthesis tool, and a final net list has been created.

This design of the PCI EXPRESS Transmitter is capable of transmitting Data at 2.5GBps. This design of the PCI EXPRESS Receiver is capable of receiving Data at 2.5GBps.The Functional-simulation has been successfully carried out with the results matching with the expected ones

The design has been synthesized using FPGA technology from Xilinx .Finally, implemented on XILINX FPGA VIRTEX Board and tested its functionality using Chip Scope Pro tool.

6 FUTURE WORK

PCI Express architecture provides smooth migration to future. PCI Express provides advanced features for the next decade of Compute & Communication platforms.

- PCI Express Server I/O Module.
- PCI Express Cable Specification.
- High End Graphics Electro Mechanical Specification.
- Advanced Switching Specification.
- Enhanced configuration of 4Kb/device.
- New Card connector.
- Server Input and Output Module (SIOM).
- Native Hot Plug support for new form factors and modules Express Card.

PCI Express enables new usage models Isochronous support for streaming media for TV Tuners, graphics, camera. In the future, PCI Express communication frequencies are expected to double and quadrupled to 5Gbps and 10Gbps .Taking advantage of these frequencies will require Physical Layer re-design of a device with no changes necessary to the higher layers of the device design.

7 REFERENCES

[1]. PCI Express Base Specification Revision 1.1, PCI-SIG, March 28, 2003. http://www.pcisig.com/specifications/pciexpress/base

[2]. PCI Express System Architecture, Mind share, Inc. 2003

[3]. PCI Express System Architecture Book, First Edition, First Printing, September 2003.

[4]. PCI Express System Architecture Book, First Edition, Second Printing, December 2003.

[5]. Intel Developer Website http://www.intel.com/technology

[6]. Michael John Sebastian Smith, “Application-Specific Integrated Circuits”, Published By Pearson Education (singapore) pvt.Ltd.

[7]. Milos Ercegovac, Tomas Lang “Introduction to Digital Systems”, Published by John Wiley & Sons

[8]. Douglas J Smith, “HDL Chip Design”, Done publications Fourth Edition 2002.

[9]. Synthesis and Verification Design Guide: http://xillinx.com/.

[10]. Elastic Buffer Implementations in PCI Express Devices, MindShare, Inc., Joe Winkles , November 2003.