International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Design and Implementation of Optimized PCI Express Physical Layer for High-Speed and Low-Latency Data Transfer

Rangaswamy B K[1], Keshava A[2]

[1]PG Scholar, *Department of ECE, BIT* Bangalore, India
[2]Assistant Professor, *Department of ECE, BIT*, Bangalore, India

Abstract
The PCIe (Peripheral Component Interconnect Express) Protocol is crucial for establishing high-speed data communication between computer peripherals, such as graphics cards and network cards etc. This communication protocol is transmitting data in packet format, with each packet containing both data and destination address and other essential information for accurate data delivery. This paper focused on Physical Layer to achieve High-Speed Data Transmission by reducing delay parameter. To minimize disturbances and enhance reliability, the physical layer uses scrambling technique and an 8b – 10b encoding technique is used for synchronization and error detection. Additionally, SIPO and PISO converts data format to improve efficiency and accuracy. The design is implemented using the Cadence compiler targeting 45nm process technology. This design is delay efficient resulting in path delay of 5.0ns and the operating frequency of 200MHz, power consumption of 1.2mw and an area of 1999µm².

Keywords: PCIe, Scrambler and Descrambler, 8b-10b encoder and 10b-8b decoder, Packets, PISO and SIPO.

I. INTRODUCTION

PCIe (Peripheral Component Interconnect Express) is a high-speed serial expansion bus for computer that displaced the parallel ones like PCI-PCI-X and AGP which are not in use presently in computers. Consequently, it has features as speed, scalability & performance improvements PCIe is based on the point-to-point connection where each device is connected directly to an end node, most frequently the CPU or chipset. This dispels the contention that comes along with the shared bus designs, and thus enhances the performance of the overall system, and comes with reduced latency. Point-to-point connections are mostly packet based and Quality of service (QoS) enable low latency. These features are critical for any application that requires to load and transfer large volumes of data such as in gaming, computes intensive applications HPC or Business Intelligence and Analytics applications. Increasing system reliability is possible with the help of hot-swap, error detection and correction (EDC) features used within the context of the PCIe interface.

PCIe is a serial, point to point, and packet-based protocol which is maintains and defined by the PCI-SIG (PCI Special Interest Group). PCIe is actually a replacement of the PCI which is a chip parallel bus protocol. PCIe is a high-speed serial; this is the main reason for choosing PCIe to support the high-speed transmission of data for this machine computer expansion bus standard used for the connection of

peripheral devices to a motherboard. Therefore, the user mode granularity of the PCIe protocol offers a point-to-point fast connection of devices such as Graphic Cards, sound cards, Networking cards and Storage devices [6]. It is seen from Fig.1 that the architecture of PCI Express is one of the most complex forms of connection over the Peripheral buses.

PCIe is divided into three layers: Transaction Layer and Data some of which are the Data Link Layer commonly referred to as the Link Layer, and the Physical Layer [3]. Based on the direction of data flow each of these layers is sub divided into two sections. The outbound section is responsible for processing and transmitting data from transmitting device to receiving device, while the inbound is responsible for receiving and processing data from receiver to transmitting device.

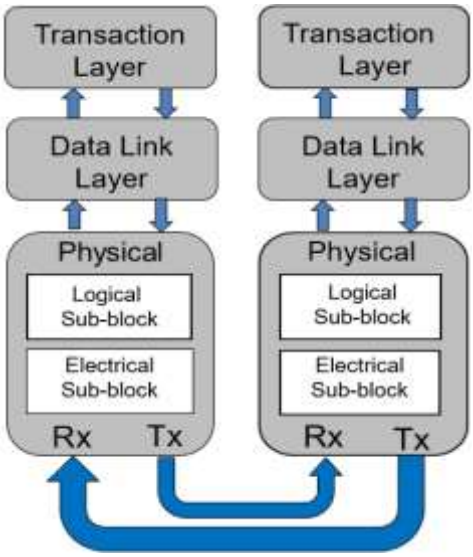

Fig 1: PCIe Packet Flow

In the case of PCI Express, packets share information between linked devices and they are created in two layers; the Transaction Layer and the Data Link Layer. As these packets pass through these layers, they are chocked with more information to aid in their handling. On the receiving device this information is then stripped off to get the actual sensed data. It is a packet-based technique that offers a high chance to use band and ensure the connivance of equipment's at a specific time. The Transaction Layer is still located as the uppermost layer with relation to software. This layer is also concerned with creating and ending of Transaction Layer Packets (TLPs). These are used during read and write operations of data and any other actions that occur in the channel between TLPs. The Data link Layer is also used for CRC (Cyclic Redundancy Check) for security purpose.

Physical Layer revolves around the basic structure of the entire PCI Express architecture with all the circuits required for interface operations integrated into this layer. It's split into two parts: It was then deemed to be the logical layer and it was also given the name of the electrical layer. It is also tasked for receiving packet from the Data Link Layer to serialize the logical layer data stream. These steps include data encoding and decoding by means of the 8b/10b, on the one hand, and data scrambling and des scrambling, on the other hand, as well as parallel to serial or serial to parallel conversion as shown in Fig 2[4]. However, the electrical layer is targeted to connect to the link while the link layer is engaged in a different way. It covers the analog portion of the differential drivers and receivers required to communicate the PCI Express device to the network.

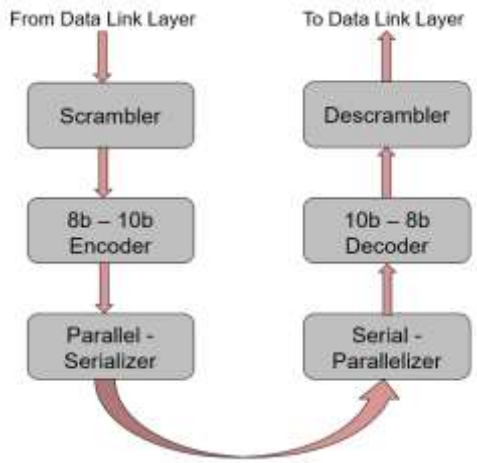

Fig 2: Data Flow Diagram of PCIe Physical Layer

The Physical Layer consists of essential ones, such as Phase-Locked Loops (PLLs) and impedance matching circuits. It remains crucial for guaranteeing efficient and fast data transfer within devices in the PCI Express communication. Further improvements to this layer, that can include individuals such as increased speed and intricate encoding, occur frequently [3]. This paper will mainly explore on the Logical Layer that is found in the Physical Layer and for this, the latest design techniques are used for this module. In PCI Express (PCIe), data communication between devices is carried out through packets formed across multiple layers of the PCIe architecture: below the Transport Layer it has the Transaction Layer, Data Link Layer, and the Physical Layer. Starting at Transaction Layer, Transaction Layer Packets, referred to as TLPs, are initiated in order to encase the data and type of transaction. These TLPs consist of the header which may contain the address, ID of the requester, and type of command, and data portion if needed and ECRC bit for check sum as shown in Fig 3. The TLP, therefore, when formed, passes to the Data Link Layer and is encapsulated into a Data Link Layer Packet (DLLP). This layer puts a sequence number with a Data Link Layer CRC (LCRC) for data check, controls flow of data and performs retransmissions in case of errors. Last of all, the packet gets to the Physical Layer, and it gets into the frame for transmission in the PCIe channel. This entails converting the data into a format that is less inclined to pick up electromagnetic interference before encoding it into a form which is acceptable such as 8b/10b format before converting it from parallel to serial for transmission. Such a layered structure helps in effective and successful transmission of data between the devices.

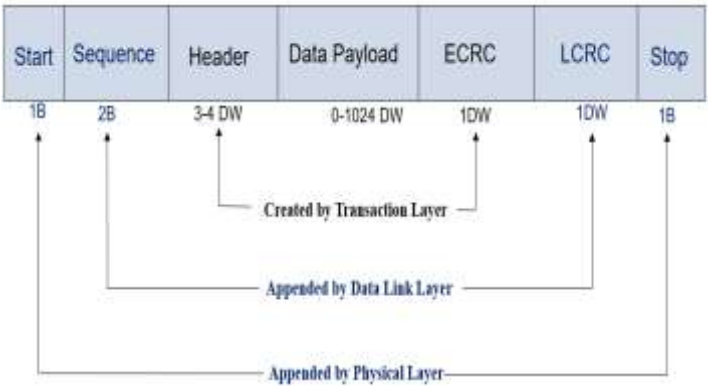

Fig 3: PCIe Packet Frame

II. SCRAMBLER/DESCRAMBLER

Form past decade, the improvement in the electronics industry is evident; mobile phones, personal computers, and satellite communication. Every electronic communication entails the transfer of data and if this data consists of large units transmitted at high frequency, they are vulnerable to Electromagnetic Interference (EMI). EMI refers to interference that affects the functionality of the circuits in electronics, hence being unwanted. For EMI, there is scrambler that plays a role in combating the problem. EMI noise is commonly present in several data streams which are characterized by repetitive patters such as "0101010101." These patters have a high density of energy which causes interference. A scrambler eliminates these patters, the overall density of energy is spread over a wider range hence minimizing interference. At the transmitter side before the data is encoded, there is some scrambling done on the data. The descrambling of the received data takes place after decoding at receiver end of the data transmission.

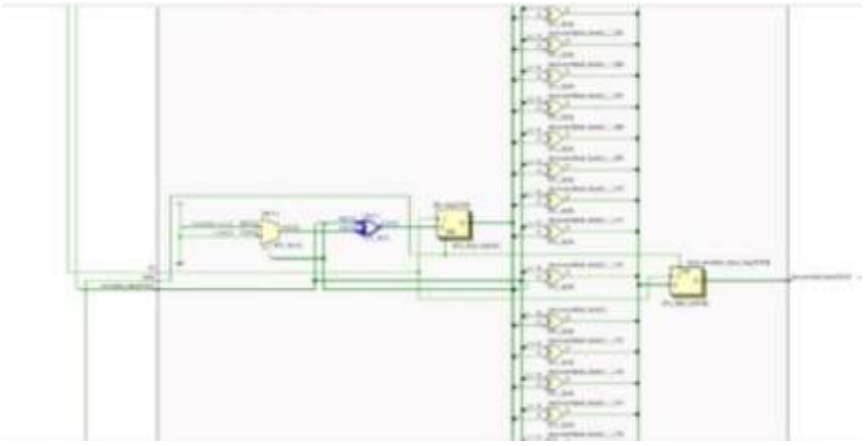

Fig 4: Schematic of Descrambler

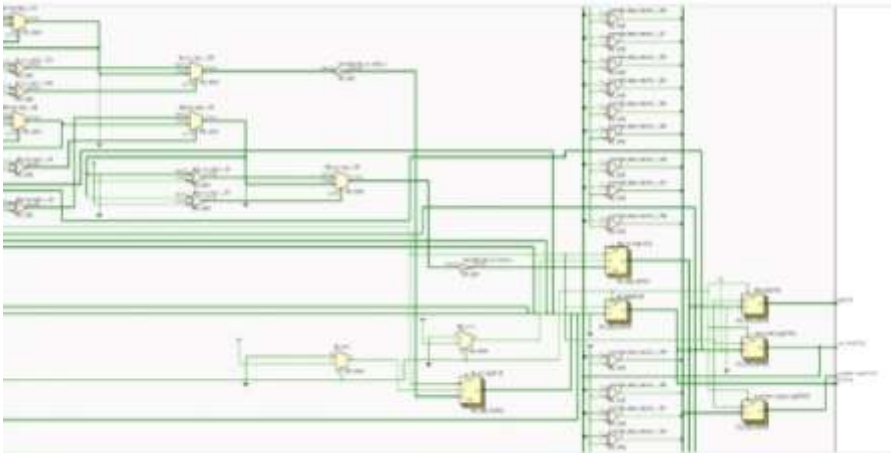

Fig 4: Schematic of Scrambler

There is a certain polynomial of LFSRs used for designing of scramblers. An LFSR has one or more flip flops in which a binary sequence is stored. It is an initial value and then in every clock cycle the proper values shift left or right. It is the feedback that decides which portions have to be XORed with each other to get the next value in the series using the polynomial. In this regard, while the LFSR makes use of the polynomial G(X) = x^23 + x^5 + 1 with an initial value of 7FFFFF. This results 8-bit data received from upper layer called Data Link Layer, being XORed with the output of LFSR.

Fig 6: Scrambler simulation result

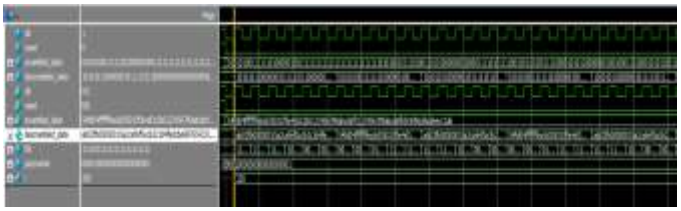

Fig 7: Descrambler simulation result

At the receiving end, the descrambler gets the scrambled output and then it descrambles it and gets to the 8 bits of data. The RTL design of the scrambler and descrambler shown at the Figures 4 and 5, and at the Figure 6 and 7 there is the simulation result.

III. 8B-10B ENCODER/DECODER

The 8b/10b encoder/decoder is actually critical for effects of security in the relaying of data. It actually receives 8-bit data and transform it into 10-bits code format. At the other end, the data is converted back to this code from the original code used in the sending end [2]. This process is mainly focused because in serial data transmission, there is no separate clock line to give a timing signal, variations in the time taken to transmit the data and the time taken to receive the data may be a problem.

The 8b/10b method has two main benefits: First, it puts a condition that each digit one and zero should be transmitted for the same amount of time or in equal small intervals so that the transmission line is balanced. Second, the property restricts the emission of the strings 1 or 0 more than 5 times, which helps the receiver to understand when the data bits are being sent. These features assist in keeping the quality of the signal with extra assurance that the entire system functions effectively.

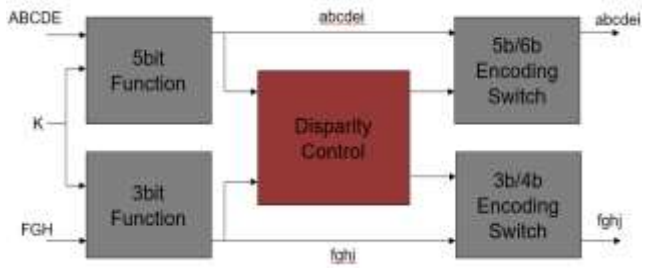

Fig 8: Design model of 8b – 10b encoder

To correct the imbalance and guarantee a balanced stream of 1's and 0's the Running Disparity (RD) is computed. Thus, when the GOP is encoded, the complexity is balanced by modifying the RD value as required. First of all, RD is assumed to be negative (RD-). Namely the count of 1s and 0s alters the RD value, which can be illustrated by a specific Fig 8.

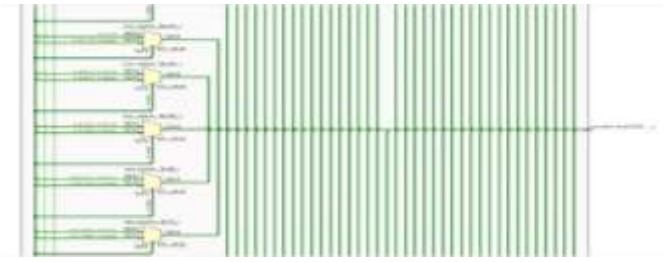

Fig 9: Schematic of 8b – 10b Encoder

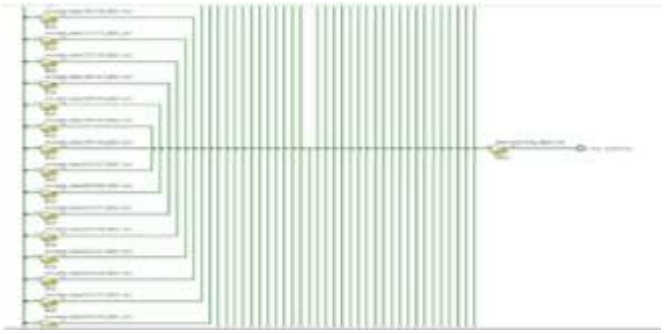

Fig 10: Schematic of 10b – 8b Decoder

The complete structure of the 8b/10b encoder is shown in yet another figure that is the figure 7. The method begins with the division of the 8-bits input data into the 3-bit and the 5-bit data. According to the RD value estimated in the current moment, the decision on the data supplement is made. Next 3-bit and 5-bit segments are encoded to 4-bit and 6-bit segments, separately. These encoded parts are then summed up to, produce a 10-bit data steam as shown below.

The decoding process simply does the opposite of this in which it extracts the 8-bit data from the 10-bit encoded data. As in the mentioned paper [1], the encoding and decoding processes apply a definite coding table that should be used. The RTL of the 8b/10b encoder and decoder are illustrated in the Fig 9 and Fig 10. On the other hand, the simulation result of the integrated circuit can be observed in Fig 11.

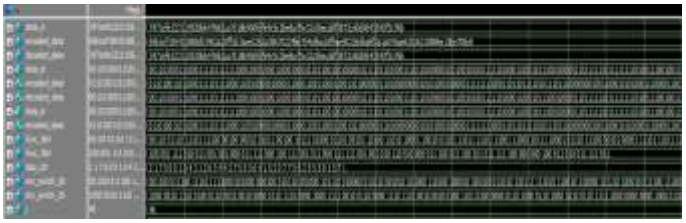

Fig 11: Simulation results of 8b-10b encoder/Decoder

IV. PARALLEL-TO-SERIAL/SERIAL TO PARALLEL

SIPO and PISO are the two important shift registers that used in building up the PCIe system. SIPO registers are used to take in the serial data transmitted over the PCIe lanes that it received and decode it into parallel data that can be processed by the internal I/O. PCIe guarantees serial and high-speed transmission of data, however, other subsystems such as processors and memory interface work on parallel data. Due to the conversion of this serial data to a parallel type of data, these components are easily manageable by SIPO shift registers.

On the other hand, PISO registers convert parallel data of the internal elements towards series data to be

transmitted back across the PCIe channels. Blocks inside the system produce parallel data that has to be followed by serialization to achieve high speed over PCIE channels. Data conversion for the purpose of its fast and efficient transmission is managed by PISO shift registers. Triggered by the rising clock signal for SIPO, the data stored in the register is transferred to the data output line while PISO transports the data stored in the register into the parallel data format, thus enabling a return of proper data conversion for PCIe systems.

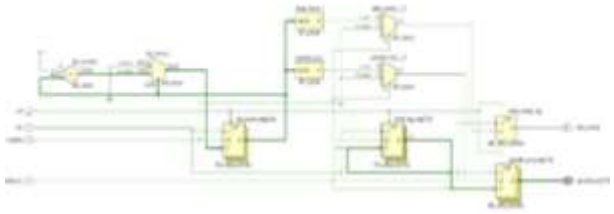

Fig 12: Schematic of SIPO

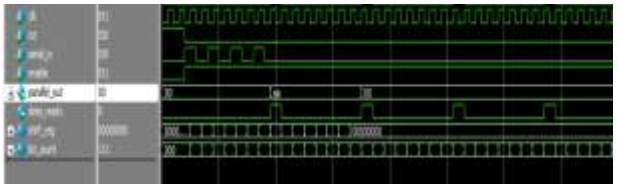

Fig 13: Simulation result of SIPO

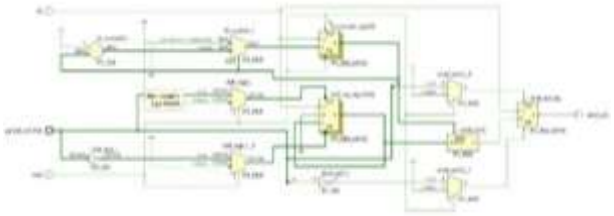

Fig 14: Schematic of PISO

Fig 15: Simulation result of PISO

V. SIMULATION RESULT

The modules of PCIe physical layer protocol are design in Verilog and simulation result is verified on Cadence. This PCIe protocol supports 192 bits for transmission. The simulation results of individual modules are shown in above figures. The transmitter module consists of Scrambler, 8b - 10b Encoder and Parallel – Serializer and the obtained result from transmitter is a serial data stream: 001010010010111010110111001100010100111001100001001101100100110010000010101011011001 1010010110001.

This serial data is fed into receiver block, which includes Serializer – Parallelizer, Decoder and

Descrambler. The final result from receiver matches the transmitted data, is: ab02fb00000015a2cafef0a1b2c3d4fedcba9876543210fe. The Fig 16 illustrates the schematic of transmitter and receiver and Fig 17 illustrates the simulation results of PCIe Physical layer Protocol

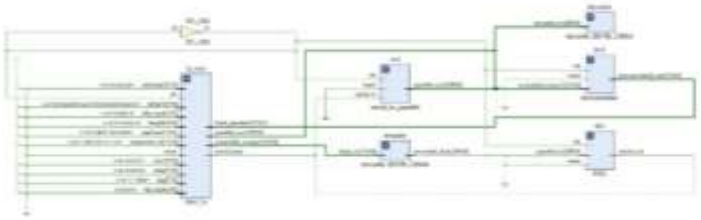

Fig 16: Schematic Diagram of PCIe Physical layer

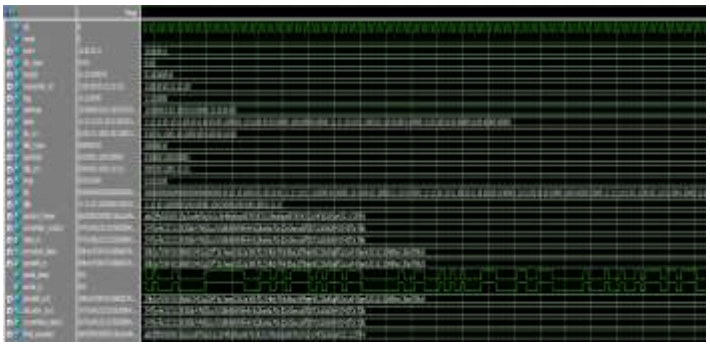

Fig 17: Simulation wave form of PCIe Physical layer

The synthesis reports are tabulated below and compared with conventional design in Table 1:

Table 1: Comparison table of Area, Power, Timing report and Frequency with conventional design.

Parameters	Conventional Work (Ref no. 01)	Proposed Work
Area (µm²)	700	1999
Power (w)	64.41µ	1.2m
Timing (ns)	9.4	5.0
Frequency (MHz)	106.3	200

VI. CONCLUSION

The Physical Layer of PCIe protocol has been implemented using Verilog and its simulation results have been validated on the Cadence platform. In this design, data transmission occurs in the form of packet, where each packet contains both the data and the destination address. This proposed design demonstrates significant efficiency in terms of delay, achieving an arrival time of 5.0 nanoseconds. Additionally, the design is optimized for resource utilization, requiring 1999 square micrometers of area and consuming 1.2 milliwatts of power when operating at a frequency of 200 MHz for data transmission.

VII. FUTURE SCOPE

For future scope, this will be implemented for multiple lanes and use 128b to 130 encoding for high security and balancing the data. Implement this design on FPGA board for synthesis and verify the design,

which transmit and receive data is same. Design the whole PCIe Protocol includes Transaction Layer, Data Link Layer and Physical Layer for application use.

REFERENCES

1. Pushpanjali J, “Implementation of PCIe Using Lower-node Technology”, 2021 International Journal of Engineering Research and Applications.
2. S. Hukare, V. Vyas and V. Agrawal, "Design and Simulation of Physical Layer of Peripheral Component Interconnect Express (PCIe) Protocol," *2023 4th International Conference for Emerging Technology (INCET)*, Belgaum, India, 2023, pp. 1-4, doi: 10.1109/INCET57972.2023.10170286.
3. S. R. Mantripragada and P. Mopuri, "Verifying performance of PCI express in a system for multi giga byte per second data transmission," 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, 2016.
4. E. Meng and X. Bu, "Design and Implementation of High-Speed Transmission Link Based on PCI-E," 2020 Information Communication Technologies Conference (ICTC), Nanjing, China, 2020.
5. H. Nakamura, H. Takayama, Y. Yamaguchi and T. Boku, "Thorough analysis of PCIe communication," 2017 International Conference on Reconfigurable Computing and FPGAs (Reconfigure), Cancun, 2017.
6. Won Ok Kwon, kyoung Park, myung Joon Kim, Daejem, “Apparatus and Method for scrambling/descrambling 16 bit data at PCI Express Protocol.”,United States patent Application Publication.
7. Si-liang Hua, Dong-hui Wang,“A 1.1 GHz 8B/10B encoder and decoder design,” 2010 Asia Pacific PCI Conference on Postgraduate Research in Microelectronics and Electronics,” pp. 138141, Nov. 22-24 Sept. 2010.
8. “Design and Verification of Peripheral Component Interconnect Express (PCIe) 3.0”, K U Prasad Bhat,Hrishikesh Ravish, Anirudha V, Prabhavathi P, 2020.
9. Chih-Hsien Lin, Chih-Ning Chen, You-Jiun Wang, Ju-Yuan Hsiao and Shyh-Jye Jou,” Parallel scrambler for high-speed applications,” in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 53, no. 7, pp. 558-562, July 2006.
10. “Universal Serial Bus 3.0 Specification. Revision 1.0”, “Appendix A,”November 12, 2008.
11. Chen Mengjie, Yu Haixun. “Module design for 8B/10B code in fiberchannel,” Northwestern Polytechnic University. ElectronicMeasurement Technology vol.30 May. 2007.
12. Wen Long, Fan Xiao-ya. “8B/10B Decoder Design,” Science Technology and Engineering. Vol.7, No.18, Sep.2007.
13. Yong-woo Kim, Jin-ku Kang, “An 8B/10B Encoder with A ModifiedCoding Table”, IEEE Asia Pacific Conference of Circuits and Systms, APPCCAS, 2008.
14. Q. Wang, S. -l. Hua and D. -h. Wang, "A 1.1 GHz 8B/10B encoder and decoder design," 2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (Prime Asia), Shanghai, China, 2010.
15. Peter Bohn, “Incremental and Verified Modelling of the PCI Express protocol” in COMPUTER-AIDED DESIGNOF INTEGRATED CIRCUITS AND SYSTEMS, OCTOBER 2010.
16. “PCI Express Base Specification Revision 1.0”, by PCI-SIG, 2002. R. Budruk, D. Anderson and T. Shanley, “PCI Express System Architecture”. MindShare Inc. Boston, MA: Addison Wesley, 2003.
17. Albert X. Widmer and Peter A. Fransazek, “A DC-balanced, PartitionedBlock, 8B/10B transmission

Code," IBM J. Res. Develop, vol. 27, pp.440-451, September, 1983.