Design and Implementing a PCI Express Serdes Block Using HDL

Ravali Meesa(✉) and G. Surekha

GRIET, VLSI, Hyderabad, India
meesa.ravali14@gmail.com, surekha537@grietcollege.com

Abstract. This paper introduces a proposal for implementing the Physical Link Layer of PCI Express in accordance with PCI Express 2.0 standards. PCI Express, a high-performance, point-to-point communication protocol, has revolutionized the world of data transfer by offering exceptional bandwidth, making it the go-to choose for a wide array of applications. Its layered architecture, consisting of three distinct layers, facilitates efficient data transfer through packet-based communication between the layers. Widespread use in various applications, including high-speed storage devices, graphics cards, and network cards. The work encompasses the design and verification of multiple physical layer blocks for PCI Express. These blocks are modeled using Verilog at the RTL level and verified using Questasim, a tool from Mentor Graphics.

Keywords: PCI Express · Physical layer (PL) · Physical Coding Sublayer (PCS) · PCI · PCIx

1 Introduction

1.1 Generation Buses

Buses are used to connect several components of a system, including the CPU, main memory, and I/O devices. Buses commonly transport data, address, and control signals [1]. Following is an explanation of how bus generations have developed:

Second-generation buses include Peripheral Component Interconnect (PCI), Advanced Graphics Port (AGP), Serial Parallel Interface (SPI), Peripheral Component Interconnect Extended, and Universal Serial Bus (USB). PCI and PCI-X are sometimes known as Parallel PCI. These second-generation buses stand out for their quick speeds, low power requirements, and economical operation.

USB 3.0, USB 3.1, SATA (Serial Advanced Technology Attachment), and PCIe (Peripheral Component Interconnect Express) are examples of the third generation of buses. [2]. These third-generation buses go from parallel to a more effective serial connection bus and employ a serial lane-based design. These third-generation I/O buses have great performance and are often used in a range of gadgets, including portable electronics, desktop computers, servers, and workstations, for a variety of computing and communication applications [3].

X. Cheng (Ed): BROADNETS 2024, LNICST 601, pp. 291–301, 2025.
https://doi.org/10.1007/978-3-031-81168-5_27

1.2 Evolution PCI Express

PCIe, is a serial expansion bus standard used to link numerous peripheral devices to computers for high-speed data transmission. It serves as a crucial conduit for information to go between these external devices and the internal parts of a computer, such as the RAM and CPU. Data exchange inside a computer system is streamlined via PCIe, which provides standardized interfaces for effective communication [4].

This standard, which is an improved version of PCI-X (Extended), was developed cooperatively by Intel, Dell, HP, and IBM. Depending on how it is implemented on the motherboard, PCIe setups can include 1, 4, 8, 12, 16, or 32 lanes. Notably, PCIe replaces prior bus technologies for computers, including the Accelerated Graphics Port (AGP), PCI-X, and the original PCI. In order to emphasise PCIe‘s benefits and improvements in the areas of data transmission and peripheral connection, this article does a comparative study, compares it with various other bus standards [5].

The choice of a serial format for interfaces like PCIe is driven by the need to minimize timing skew in comparison to parallel formats. Timing skew, a consequence of finite signal speed, arises when various traces within an interface possess distinct lengths, causing parallel signals from the source to reach their destination at different intervals. This variation in arrival times makes it challenging to correctly capture parallel word bits at the destination. Maximum bandwidth is attained when the speed of the data aligns with the differences in distance between parallel signal channels that are long and short. A few well-known serial interconnects are FireWire, USB, SATA, and RapidIO. To accommodate devices with varying speeds, serial multichannel representation allocates fewer lanes to slower devices, thereby enhancing flexibility in data transfer[1, 5].

2 PCI Express Layers

The Transaction Layer (TL), data link layer, and physical layer are the three levels that the PCI Express uses to specify a layered architecture flow is show in Fig. 1.

2.1 Transaction Layer

Transaction Layer Packets (TLPs) are the name for packets that are entering the TL. These TLPs are made up of a header, data, and, if desired, ECRC fields that provide end-to-end data integrity checks.Outbound TLP traffic is produced within the TL, and inbound TLP traffic is likewise received and handled[6, 8].

TLP are the name for packets that are entering the TL. These TLPs are made up of a header, data, and, if desired, ECRC fields that provide end-to-end data integrity checks. Outbound TLP traffic is produced within the TL, and inbound TLP traffic is likewise received and handled[7, 12].

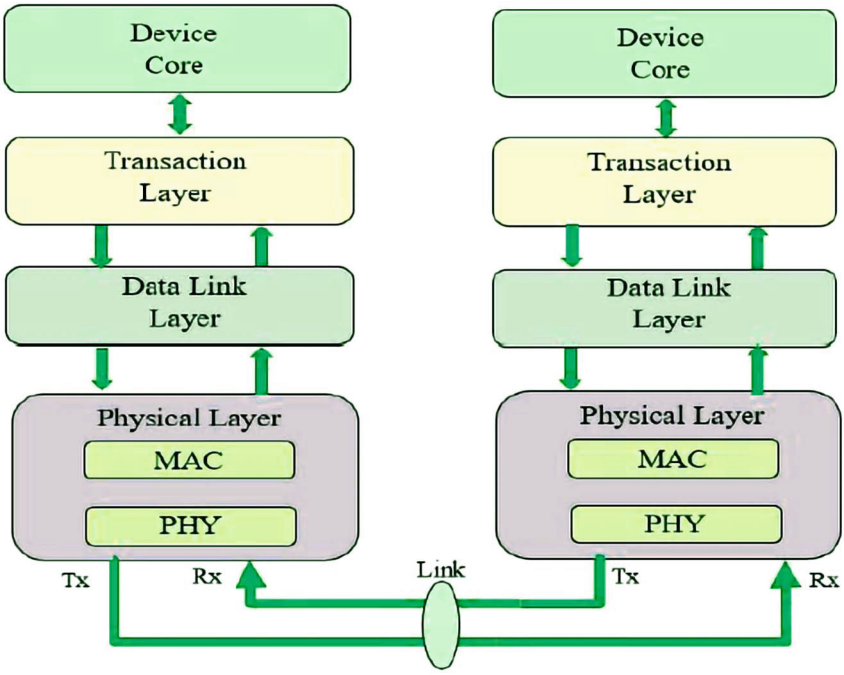


Fig. 1. The transfer of data between PCI Express devices and their layer

2.2 Data Link Layer

Sequence numbers and a link-level CRC (LCRC) are added to incoming TLPs from the TL in the data link layer (DL). These sequence numbers are essential for ensuring that packets are sent successfully. The DL layer is in charge of ensuring efficient power management, as well as facilitating reliable data exchange, error detection via a 32-bit Cyclic Redundancy Code (CRC), implementing an acknowledgment protocol (ACK/NACK signaling), managing retries, initialising and updating flow control credit (FCC), and managing retries. The DL creates and manages data link layer packets (DLLPs) to carry out these tasks[9].

2.3 Physical Layer

PHY, short for Physical Layer Interface, stands for the OSI model‘s physical layer and is necessary for carrying out physical layer operations in a network. It acts as the interface between any physical media, such as copper wires or fiber optics, and the link layer and MAC (Media Access Control) [10]. The PCS and the Physical Medium Attachment layer (PMA), which together make up the PHY component, serve as a vital interface between data link protocols and the actual physical transmission media in a network architecture as show in Fig. 2 [13].

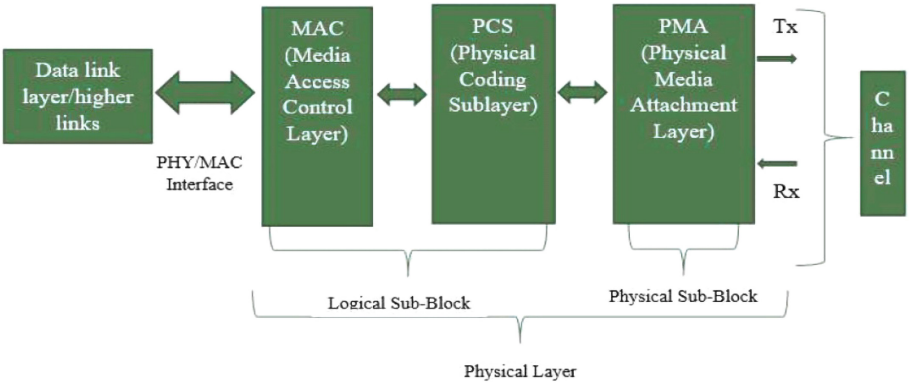


Fig. 2. Block segmentation for PCIe in the physical layer

3 Related Work

The PCIe physical layer, which is in charge of handling data transmission and reception between PCIe devices and the motherboard, includes the Physical Cod-ing Sublayer (PCS) as a fundamental component. Within the logical and electrical building blocks of the PCIe physical layer, the PCS is a crucial element[11, 14]. The total 6 block flow of the signal/Data is show in Fig. 10 and block diagram result are show in Fig. 3**.**

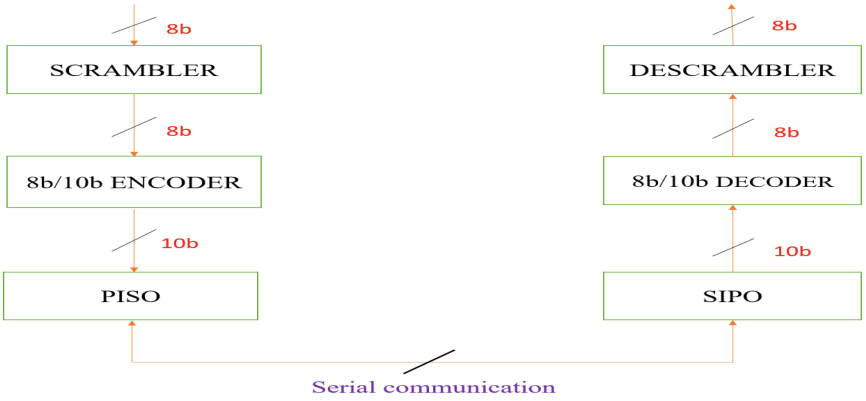


Fig. 3. Block diagram of PCIe Physical coding Sub-Layer

3.1 Scrambler

The scrambler is a vital component of the PCS of communication protocols like PCIe for enhancing data integrity and guaranteeing accurate information delivery [15]. Long sequences of identical bits are avoided by the scrambler by adding a regulated amount of unpredictability to the data stream. Such sequences may make it difficult to maintain

synchronization throughout transmission and reception, which might affect how precisely the clock signal is extracted at the receiver‘s end. With the scrambler in place, the data stream becomes more balanced and varied, enabling a higher degree of resilience against electromagnetic interference and signal distortion.

The scrambler is essential to the PCIe ecosystem‘s continued robust and high-performance data transfer between linked devices. It makes an important contribution to the overall efficacy and dependability of the PCIe standard, making it an essential element of modern computer systems and other applications that use PCIe for trustworthy and quick data transmission Fig. 4.

3.2 Descrambler

Plays an important part in data transfer by reversing the scrambling process. The descrambler ensures that the received data is synchronized and free of the effects of scrambling by removing the added randomness, preserving data integrity and preventing the appearance of extended sequences of identical bits. This synchronization and data restoration are critical for proper data recovery because they ensure that information is sent accurately and reliably across devices connected over the PCIe link Fig. 9.

3.3 8b/10b Encoder

The 8b/10b encoding scheme is a fundamental component of the PCIe standard, used to ensure reliable and efficient data transmission. 8b/10b Encoder: In PCIe, the 8b/10b encoder is responsible for converting 8-bit data words into 10-bit symbols using “look up tables” before transmission. This encoding process serves several crucial purposes. Firstly, it ensures a balanced number of 0s and 1s in the data stream, which aids in maintaining DC balance. Maintaining DC balance is essential to prevent long sequences of 0s or 1s that could cause synchronization issues or disrupt signal integrity. Secondly, it provides an element of error detection; the 8b/10b code has built-in properties that enable the receiver to identify and correct errors in the data stream. Thirdly, the 8b/10b encoding facilitates clock recovery, helping the receiver extract the clock signal from the received data Fig. 5.

3.4 8b/10b Decoder

On the receiving end, the 8b/10b decoder performs the reverse process, converting the 10-bit symbols back into the original 8-bit data words. This decoding process restores the original data while also checking for and correcting any errors that might have occurred during transmission. The 8b/10b decoder plays a crucial role in recovering the data accurately and ensuring data integrity.

In PCIe and other high-speed communication standards, 8b/10b encoding and decoding are vital for maintaining reliable and efficient data transmission. They help prevent data errors, facilitate clock recovery, and improve signal integrity, ensuring that data is successfully exchanged between PCIe devices and the motherboard with a high degree of accuracy and performance Fig. 8.

3.5 PISO (Parallel-In Serial-Out)

PISO is a mechanism used in the PCIe PCS block to convert parallel data into a serial format. It is essential for taking data from the parallel data bus within the PCIe device and converting it into a serial bit stream, which is suitable for high-speed transmission over the PCIe link. PISO is particularly useful for efficiently transmitting data between devices and the PCIe interface, ensuring it's appropriately formatted for the PCIe communication standards Fig. 6.

3.6 SIPO (Serial-In Parallel-Out)

SIPO, on the other hand, is responsible for converting the serial data received over the PCIe link back into parallel data. This is necessary on the receiving end to extract and decode the data correctly. SIPO takes the serial bit stream and converts it into parallel data words that can be processed by the PCIe device Fig. 7.

4 Result

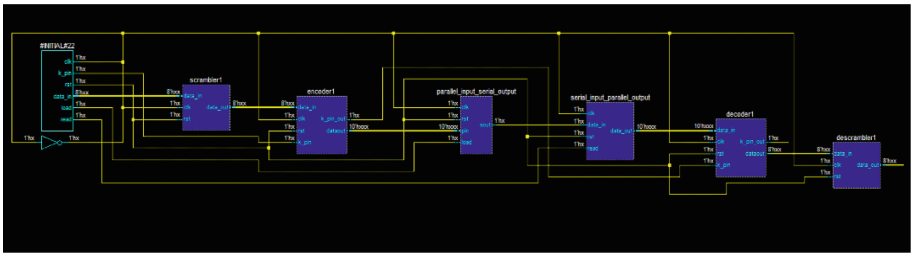


Fig. 4. Synthesize Of PCS Block

The input signal in Fig. 10 is initially processed in the scrambler block, which is the first block. The signal is then converted based on the particular logic of each block that follows. Accurate replication of the original input signal at the receiver side is essential for the PCIe PCS to work properly.

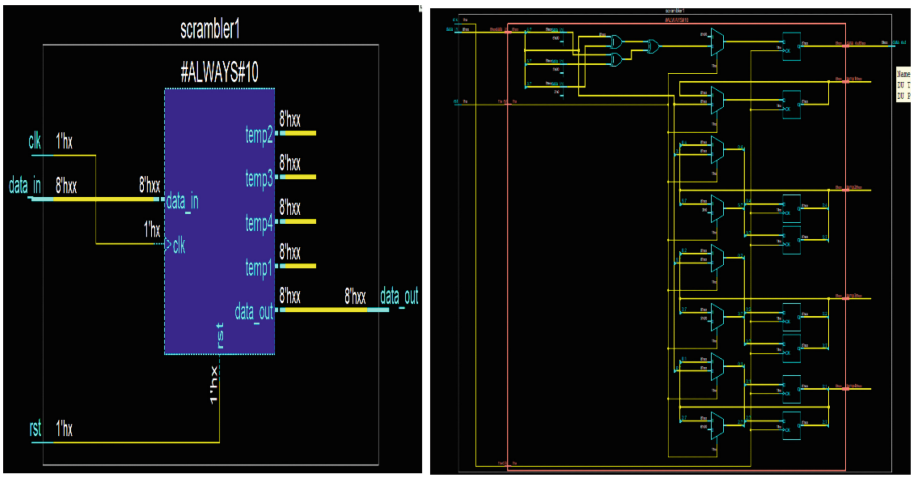


Fig. 5. Block Diagram Of Scrambler and Synthesis Of Scrambler.

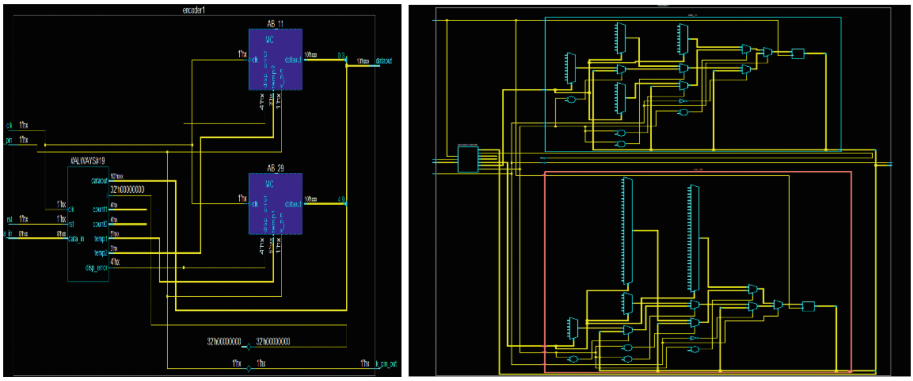


Fig. 6. Block Diagram For Encoder and Synthesis Of Encoder

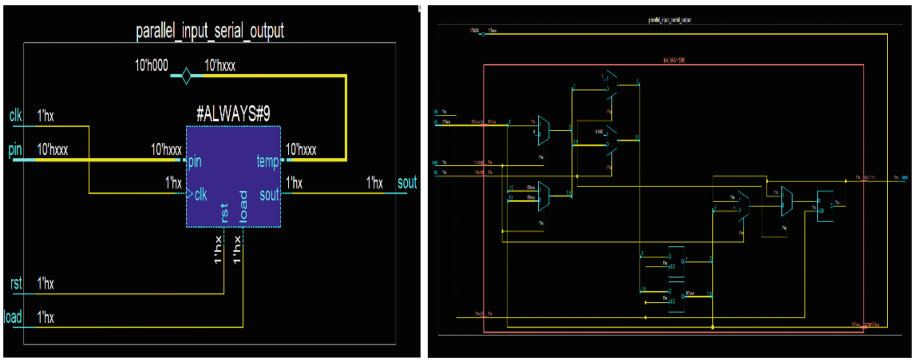


Fig. 7. Parallel In Serial Out Block and Synthesis Of PISO

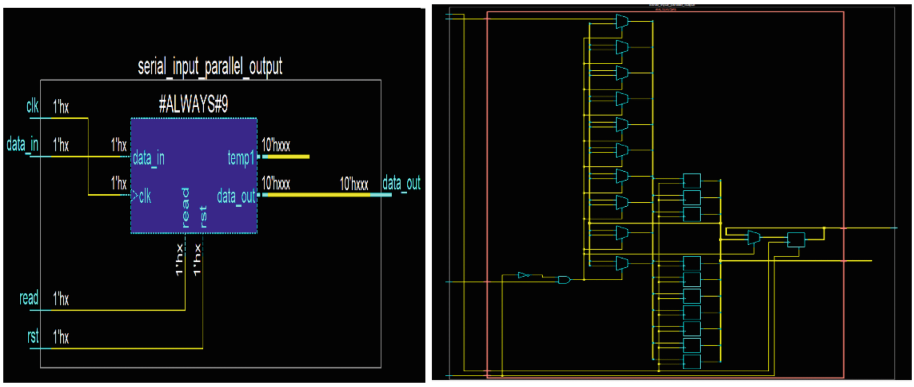
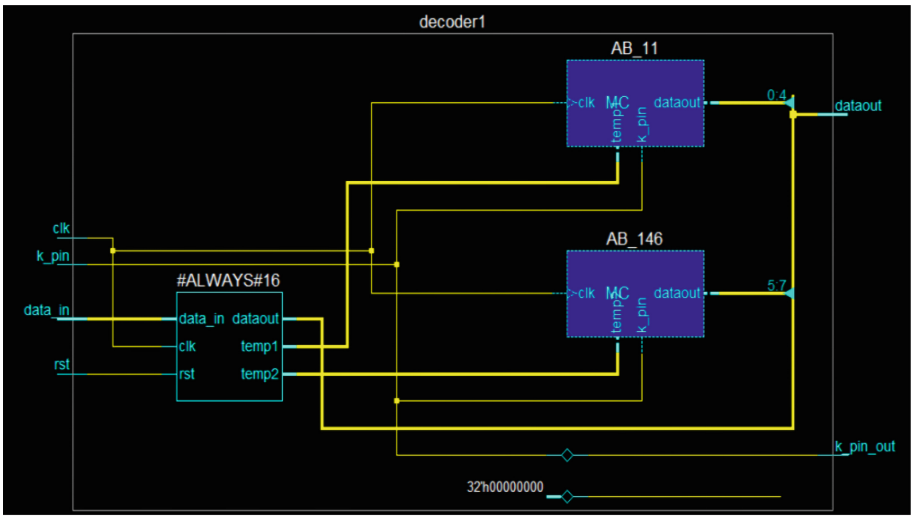
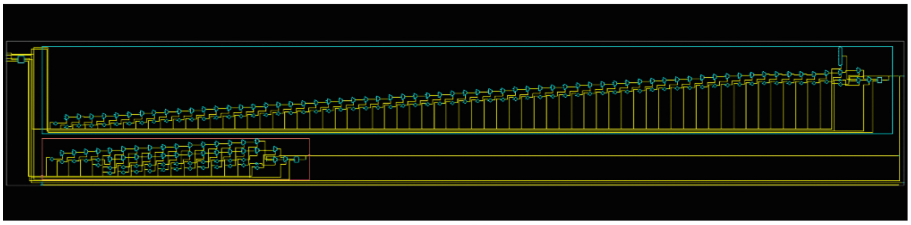


Fig. 8. SIPO Block and Serial In and Parallel Out Synthesis



(a)



(b)

Fig. 9. (a) Decoder Block. (b)Synthesis of Decoder

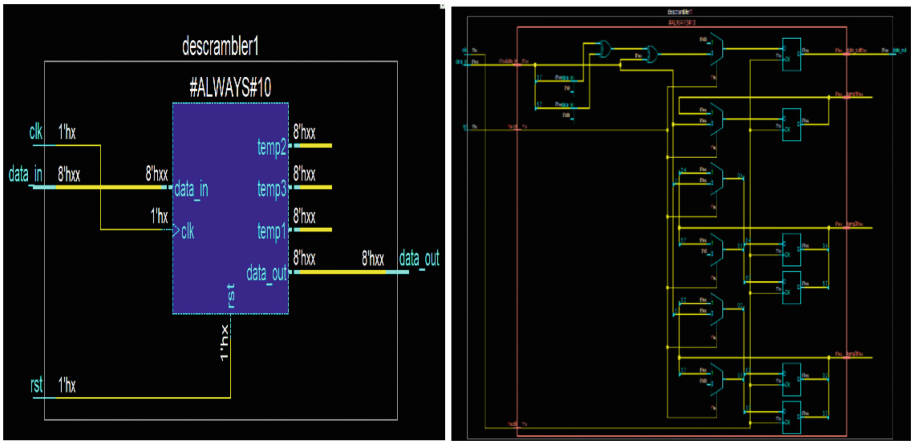


Fig. 10. (a) Block For Descrambler. (b) Synthesis of Descrambler

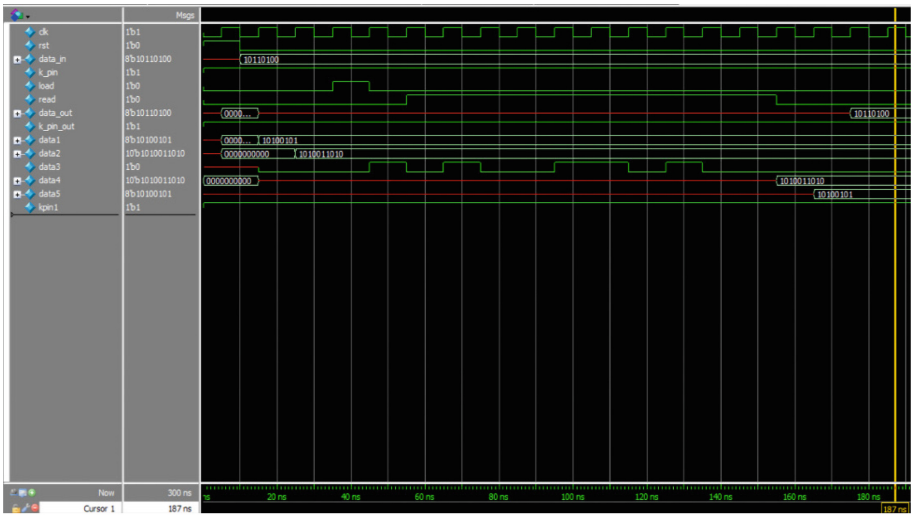


Fig. 11. Physical Coding Sublayer Total Transmission and Receiving Process Waveform

5 Conclusion

In this paper, the PCIe PCS block was meticulously examined and simulated using Questasim Tool. The research encompassed all six essential components of the PCS block: scrambler, encoder, PISO, SIPO, decoder, and descrambler. Through rigorous simulation and verification processes, it was demonstrated that the time delay for one packet to traverse from the transmitter to the receiver was remarkably low, specifically clocking in at 172 ns. This efficiency was achieved through a streamlined design approach, leveraging the simultaneous operation of key elements such as the scrambler and encoder during transmission and the decoder and descrambler during reception, all synchronized within a single clock cycle. The total power required to drive 0.068 w. By providing detailed explanations of the underlying processes and mechanisms employed in the scrambler, encoder, PISO, SIPO, decoder, and descrambler, this research has contributed valuable insights to the optimization of PCIe communication. These results hold promising implications for the future development of high-speed data transfer systems, emphasizing the importance of efficient design methodologies and robust simulations in achieving minimal time delays and ensuring reliable data transmission between devices.

References

1. Li, A.: Evaluating modern gpu interconnect: Pcie, nvlink, nv-sli, nvswitch and gpudirect. IEEE Trans. Parallel Distrib. Syst. **31**(1), 94–110 (2019)
2. Tembhare, A.R., Patil, P.B.: Design & implementation of PCI Express Bus Physical layer using VHDL. Int. J. Recent Innov. Trends Comput. Commun. **2**, 1883–1886 (2014)
3. Prasad, R.A., Kulkarni, M.: Design and verification of phy interface for pcie gen 3.0 and usb gen 3.1 using UVM methodology. Int. Res. J. Eng. Technol. (IRJET) (2017)
4. Specification, Base. PCI Express® Base Specification Revision 3.0 (2002)

5. Verma, A., Dahiya, P.K.: PCIe BUS: a state-of-the-art-review. IOSR J. VLSI Signal Proces. (IOSR-JVSP) **7**, 24–28 (2017)
6. Rangel-Patiño, F.E., et al.: PCIe Gen5 physical layer equalization tuning by using K-means clustering and gaussian process regression modeling in industrial post-silicon validation. In: 2023 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO), pp. 162–165. IEEE (2023)
7. Rohilla, G., Mathur, D., Ghanekar, U.: Functional verification of MAC-PHY layer of PCI express Gen5. 0 with PIPE interface using UVM. In: 2020 In-ternational Conference for Emerging Technology (INCET), pp. 1–5. IEEE (2020)
8. Vaidya, V.N., Ingale, V., Gokhale, A.: Development of Verifica-tion IP of Physical Layer of PCIe. In: 2022 IEEE 3rd Global Conference for Advance-ment in Technology (GCAT), pp. 1–5. IEEE (2022)
9. Liang, J.: A study of DTN for reliable data delivery from space station to ground station. Ph.D diss., Lamar University-Beaumont (2023)
10. Miryala, D.K.: Implementation of PCS of Physical Layer for PCI Express. Ph.D diss. (2009)
11. Du, L., Zhang, Z., Tong, J., Wang, C., Wang, W.: Research on XDMA high-speed data transmission architecture based on PCIe. In: 2020 IEEE 6th International Conference on Computer and Communications (ICCC), pp. 1783–1787. IEEE (2020)
12. Johansen, A.K.: Fast Multi-GPU communication over PCI Express Bench-marking PCIe transport with the NVIDIA Collective Communications Library (NCCL) using legacy GPUs. Master‘s thesis (2023)
13. Hukare, S., Vyas, V., Agrawal, V.: Design and simulation of physical layer of peripheral component interconnect express (PCIe) protocol. In: 2023 4th In-ternational Conference for Emerging Technology (INCET), pp. 1–4. IEEE (2023)
14. Dong, L.-L., et al.: Research on verification method of PCIe IP core based on programmable logic. In: International Symposium on Software Reliability, Industrial Safety, Cyber Security and Physical Protection for Nuclear Power Plant, pp. 290–301. Singapore: Springer Nature Singapore (2023)
15. Nakamura, H., Takayama, H., Yamaguchi, Y., Boku, T.: Thorough analysis of PCIe Gen3 communication. In: 2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig), pp. 1–6. IEEE (2017)