Enhancing High-Speed Data Transfer in Fpga-Based Systems: An Innovative Usb 3.0 Communication Interface Design

Cailin Liao [1], Yuhao Liu [2] and Yanzhe Zhu [3, *]

[1] School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China

[2] School of Traffic & Transportation Engineering, Central South University, Changsha 410083, China

[3] International college of Zhengzhou University, Zhengzhou 450001, China

* Corresponding Author Email: dangling@stu.zzu.edu.cn

Abstract. This study introduces an innovative design for a USB 3.0 communication interface, specifically tailored for Field-Programmable Gate Arrays (FPGAs). Given the escalating requirements for rapid data transmission in contemporary computer applications, USB 3.0 provides substantial enhancements over its antecedents. Nonetheless, extant USB communication interfaces inadequately cater to FPGA development, leading to diminished productivity for developers. To reconcile this discrepancy, this paper advocate for an Intellectual Property (IP) core interface that capitalizes on the concurrent data transmission proficiency of FPGAs, in conjunction with the superior speed transfer of USB 3.0. This design employs the CYUSB3014 chip, amalgamating the Physical Layer (PHY), Serial Interface Engine (SIE), and driver software into a singular entity. Herein, the FPGA serves as the nexus of control, streamlining data interchange and interface coordination among diverse modules within the system. The design put forth not only enables expeditious data communication in FPGA-centric systems, thereby surmounting the constraints of conventional USB interfaces, but also promotes accelerated data conveyance and augmented development efficiency. Consequently, this design harbors extensive applicability across spheres of digital system design and communications.

Keywords: FPGA, USB 3.0, high-speed communication, data transmission.

1. Introduction

A Universal Serial Bus (USB) interface is an interface that transmits data over Ethernet. Typically found in PCs designed by different companies, such as Intel, Compaq, NEC, Digital, Northern, IBM, and Microsoft [1]. In data transmission, with the growth of multimedia data such as images and videos in computer applications, the transmission rate of external device interfaces has been put forward a higher demand, and the maximum transmission speed of USB 3.0 can reach up to 5 Gb/s. The speed of USB 3.0 is faster than USB 2.0, and USB 3.0 has the advantage of lower power consumption [2-4]. The frequently used USB communication interface is not suitable for FPGA development, so developers need to design their own communication interface, resulting in lower development efficiency [5-7]. In previous research on USB communication interfaces, there have been solutions using an external chip including PHY through FPGA logic, the most common is the TUSB1310A chip from Texas Instruments, and some researchers have tried to use FPGA internal transceivers as USB-PHY to replace TUSB1310A [8]. This paper uses CYUSB3014 chip as the USB chip.

This paper uses FPGA as the main control chip and designs an interface which can use USB3.0 for high-speed communication by controlling CYUSB3014 chip. This paper mainly designs an Ip core for data transmission with CYUSB3014 chip and calls a FIFO Ip core to store the exchanged data. In this paper, USB chip firmware is designed so that FPGA can transmit data with host computer. At last, the paper simulates the designed content in time sequence to prove the correctness of the designed content. This paper makes full use of FPGA parallel data transmission and USB 3.0 high-speed data transmission and puts forward a high-speed data communication solution. It also provides

a solution for the connection between FPGA and other USB peripherals. In short, the USB 3.0 interface will become an indispensable direction for high-speed data transmission.

2. Analysis of theoretical principles

2.1. FPGA overview-defining structure and characteristics

FPGA is a programmable logic device, which is composed of a large number of logic units (Look-Up Tables, LUTs) and flip-flops. These logic cells and flip-flops can be programmed and configured according to the user's needs to achieve specific functions. The core structure consists of the following main components:

a. Logic units (Look-Up Tables, LUTs): LUTs are the most basic logic units in FPGA and are used to store and calculate logic functions. Each LUT has an input and an output that can be programmed to configure a specific logic function.

b. Flip-flop: Flip-flop is used to store and control data flow and is a storage unit in FPGA. Flip-flops can store logic states and update them under the control of a clock signal.

c. Programmable interconnect resources: Programmable interconnect resources in FPGA are used to connect logic units and flip-flops to implement various logic functions. These interconnect resources can be reconfigured programmatically, changing the way circuits are connected.

FPGA's remarkable flexibility, parallel processing capabilities and scalability have been widely used in digital system design, communications and other fields. communications and other fields.

2.2. Technical implementation of USB3.0 communication interface

FPGA acts as a control center in the entire hardware system and is responsible for constructing data synchronization of the entire system. Its main function is to realize coordination control such as data exchange and interface interaction between various modules. The FPGA interface plays an important role in this system. It must not only complete the connection with the USB controller GPIF II, but also provide a data transmission interface with the internal logic module of the FPGA. The logic of the FPGA interface is the core of the entire system. As the master device, it controls the working status of the slave device GPIF.

The system workflow includes sending control commands and uploading data. In this system, FX3 and FPGA work independently, and they use external clocks for synchronization. When sending a control command, the PC first uses the host computer software to write the command. Then, the command is sent to the FPGA through the FX3 chip. The FPGA uses the IP core to parse the command and cache it into RAM. When the system uploads data, the FPGA reads the data internally and sends the data to FX3 through the USB 3.0 interface. Finally, the data is transferred to the PC. The system's USB 3.0 chip selects CYUSB3014 from Cypress's FX3 series, which is embedded with a 32b ARM9 series microprocessor. At the same time, the chip can be connected to any ASIC and FPGA through the parallel programmable interface GPIF II. The chip is backward compatible with USB 2.0 mode. Developers The chip working mode can be set through programming. In short, the system realizes functional interaction through control commands and data transmission between FX3 and FPGA.

CYUSB3014 integrates multiple important functional modules, such as GPIF (General Programmable Interface) II interface, FIFO (First In (output) and DMA (direct memory access engine), these functional modules enable flexible data transmission and processing capabilities. Mainly used to process read and write commands. Read and write commands require the computer to control transmission through USB and pass them to CYUSB3014, which then converts the specific commands into level signals and sends them to the IO port of the FPGA. The USB interface module determines whether it is a read command, or a write command based on the level signal, and finally generates effective signals such as enable OE and read/write control of CYUSB3014.

In applications where the external processor needs to access the data buffer in the CYUSB3014 chip, synchronous slave device FIFO interface technology can be used [9-10]. The flexible use of the

synchronous slave device FIFO interface can meet the requirements of high throughput data transmission.

During the fast transmission of packet mode data, the use of the synchronous slave device FIFO interface can effectively cache data, ensure the continuity of data in packet mode, and improve the reliability index of data transmission. Therefore, in the system or module described in this article, the synchronous slave device FIFO interface design is adopted, as shown in Figure 1 [11].

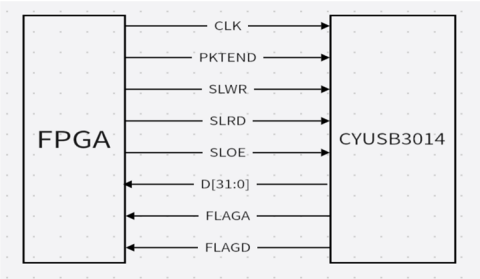

Fig. 1 Schematic diagram of synchronous slave device FIFO interface [11]

This article designs the module shown in Figure 2 based on Figure 1, in which fifo_generator is the directly called ip core, which is used to store the data transmitted by the USB3.0 communication interface. fx3_rw is the designed communication interface.

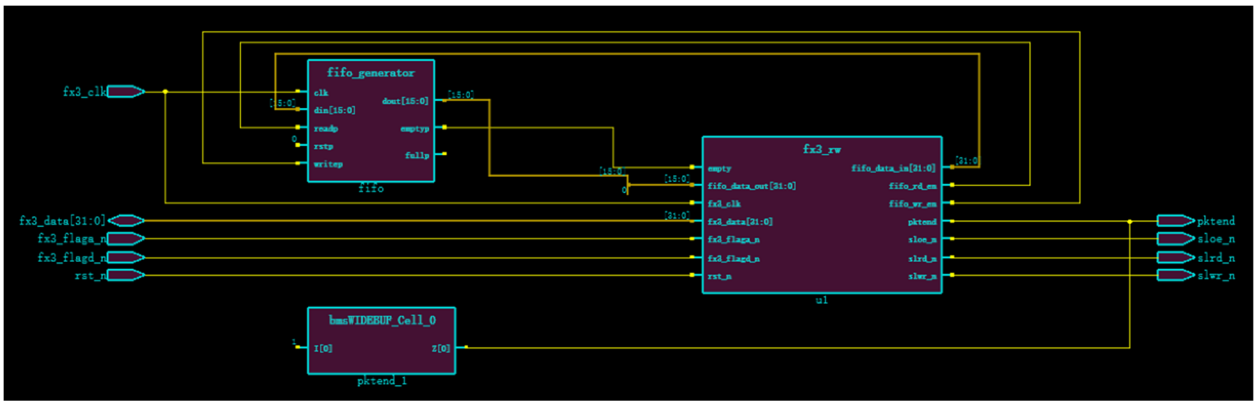

Fig. 2 Schematic diagram of overall module design (Photo/Picture credit: Original)

In addition to the necessary input and output ports, there is also a read-write state switching state machine. The specific state transition of the state machine is shown in Figure 3. After reset, it enters the IDLE state and is judged by the flag bit. Whether it is readable or writable, then it enters the corresponding read and write state, and finally the flag bit is used to determine whether it returns to the IDLE state.

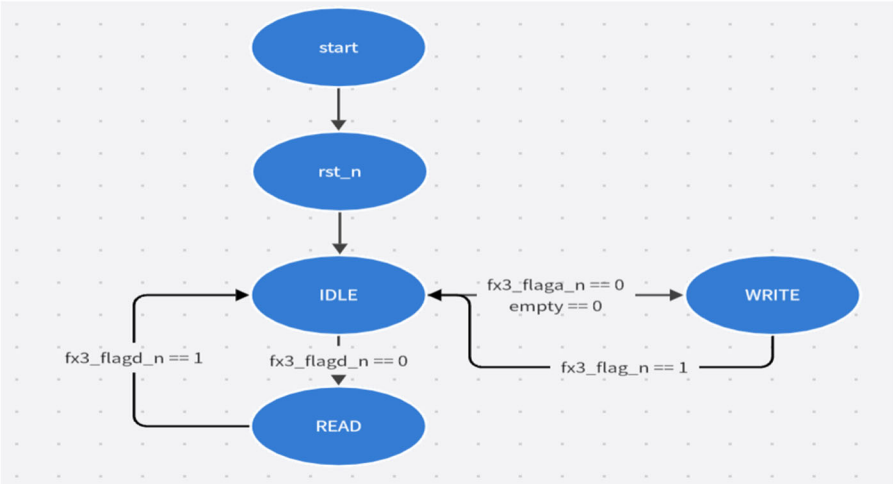

Fig. 3 Schematic diagram of reading and writing status switching (Photo/Picture credit: Original)

The timing changes of the communication interface and CYUSB3014 are designed strictly in accordance with the timing diagram shown in the data sheet, as shown in Figure 4 and Figure 5 [12-13].

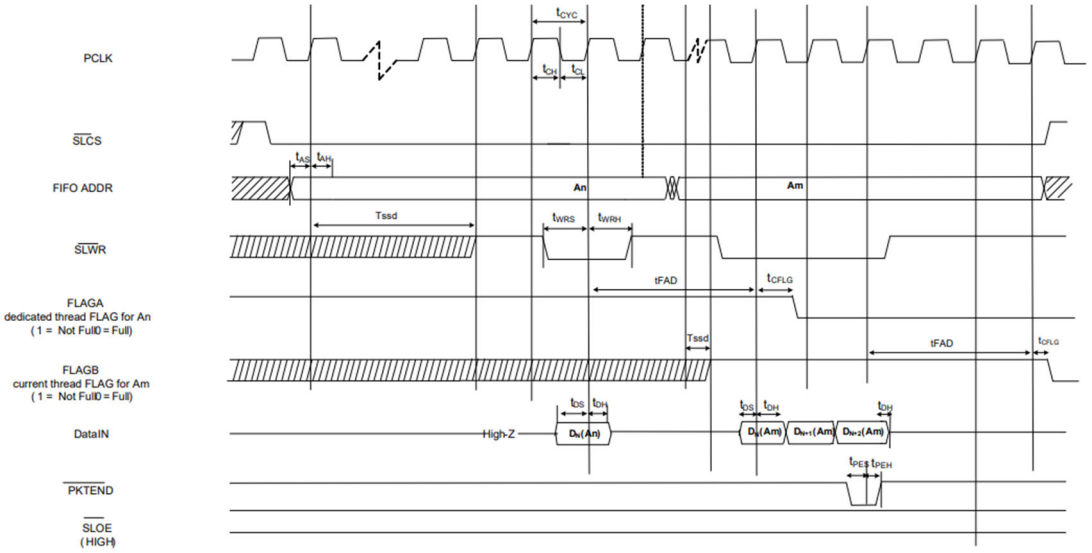

Fig. 4 Synchronous read timing [12]

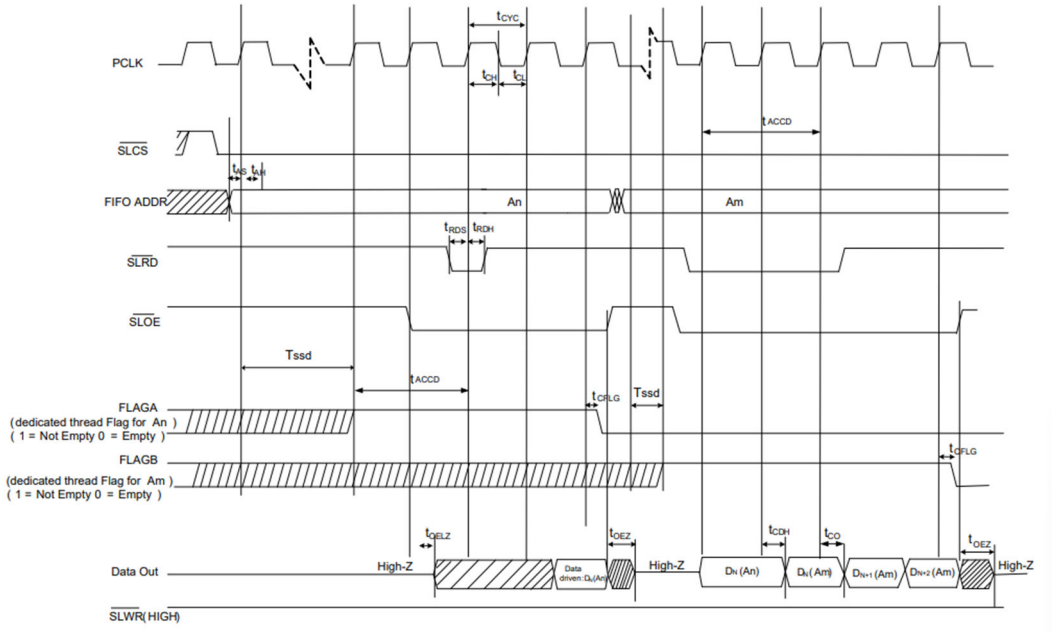

Fig. 5 Synchronous write timing [13]

3. USB3.0 firmware design

In this experiment, Slavefifo-2bit firmware provided by Cypress official is used, which mainly completes the design including chip initialization and other functions. The design of this firmware mainly included DMA channel initialization, parameter configuration, processing interrupt, etc [14]. FX3 embedded ARM9 microprocessor can be used as a bridge between the host computer and FPGA. By packaging the parallel data transmitted from the host computer into data packets with usb3.0 protocol, the most critical one is that GPIF II is the communication interface between CYUSB and FPGA, which has high performance and can theoretically achieve seamless connection with FPGA [15]. Its parameters mainly include peripheral interface, data mode and flag bit.

Modifications can be made in the visual interface, and the use of the firmware includes: (1) Set the clock module and set the system clock to 50MHZ; (2) Set up GPIF II, interface and FPGA connection;

(3) Set buffer. The maximum burst length of DMA is 16. When the burst length is 16, the larger the buffer length and range, the faster the transmission rate. But at this time, the transmission rate of small packets will decrease. Therefore, after verification, the number of buffers is set to 6, and the size is set to 16kb [16].

4. Simulation design inquiry

FX3 chip has the capability of high-speed parallel data processing, the host computer transmits commands to FX3 chip through data transmission software, and FX3 can communicate with FPGA by adjusting the function mode of GPIF II port. Considering the high-performance characteristics of GPIF II port, it is only necessary to verify the transmission function of FPGA interface. Through the above analysis, the simulation waveforms of FPGA-FIFO and Slave FIFO are verified here.

4.1. Parameter Settings

4.1.1 Slave FIFO parameters

According to the firmware design scheme, the bidirectional bus data bit width of Slave FIFO designed in this experiment is 32 bits, 4 cache areas while each cache area size is 16kb.

4.1.2 FPGA-FIFO parameters

In this paper, the FIFO in FPGA is only used for storage, parsing the data transmitted by USB, and output the data as required, without involving the change of clock domain and data bit width. So the read-able-out and write data bit width is also 32 bits, and the read-able-out and writable data depth is set to 1024, and the maximum storage of 4KB data.

4.1.3 System clock parameter

There are two internal clock modes of FX3 chip: one adopts passive external crystal oscillator, and the other introduces external system clock through CLKIN32 pin. The FPGA-FIFO and Slave FIFO in this paper share a system clock, and the internal system clock of FPGA is 50MHZ.

4.2. Simulation process and results

4.2.1 Slave FIFO simulation

Based on the experimental requirements, the Slave FIFO designed in this paper is firstly simulated and tested. Giving the clock signal a 50MHZ excitation,

For bidirectional bus data processing, this paper adopts the mode of three-state gate, when the writable flag bit signal (fx3_flaga_n) is low, the bus data is assigned a high level, otherwise it is placed at a low level [17]. Simulation results are as follows in figure 6.

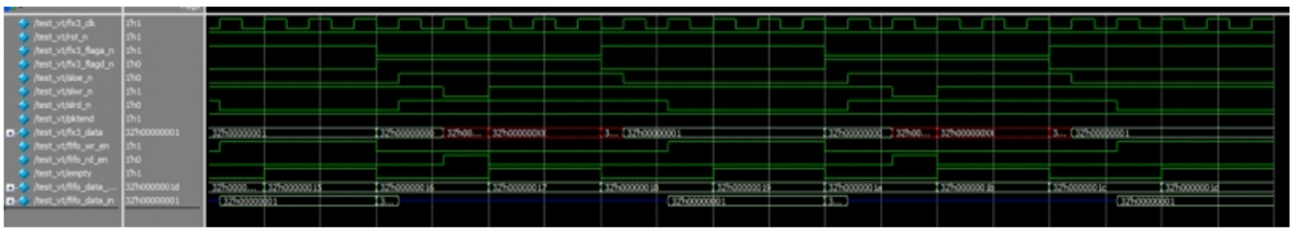

Fig. 6 Waveform diagram of Slave FIFO simulation (Photo/Picture credit: Original)

4.2.2 Top-level simulation

After the Slave FIFO module is verified successfully, the FPGA-FIFO module and Slave FIFO are packaged, and the top layer module is RTL level integrated. Figure 2 shows the synthesized circuit.

The system clock also adopts 50MHZ, and the bus data also adopts the three-state gate mode, which assigns the value of the counter when the writable flag bit signal is lowered and zeroes the rest of the time. The counter adds one to each time the clock rises. Simulation results are as follows in figure 7.

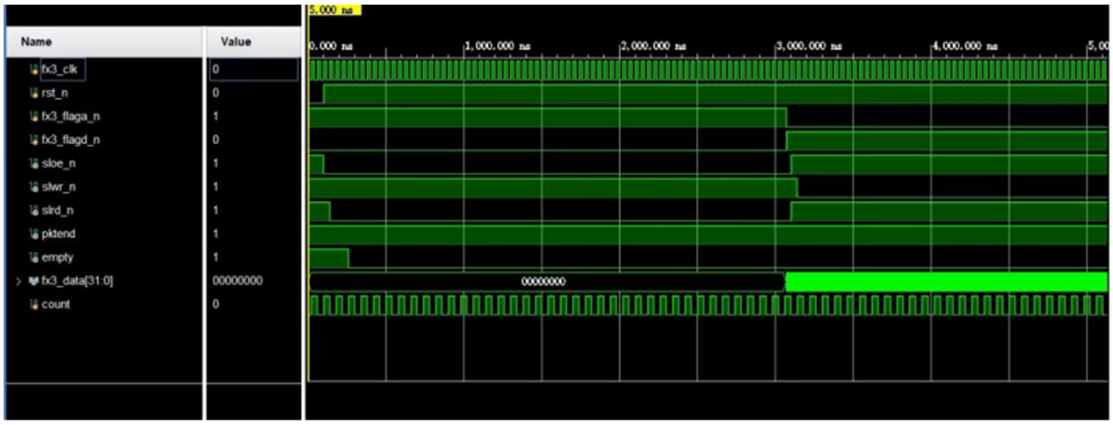

Fig. 7 Simulation waveform diagram of the top-level module (Photo/Picture credit: Original)

4.3. Result Analysis

As shown in Figure 6, when the read-out signal (fx3_flagd_n) is pulled up and the writable signal is pulled down, the read-able state is entered, the read-able enable signal (slrd_n) is pulled down, and the output bus data (fx3_data) is buffered by the FIFO module of the FPGA and finally stored in RAM. When the writable signal is high and the read-out signal is low, if the space-time signal is also low, the writable enable signal (slwr_n) is low and enters the writable data mode. At this time, the data stored in the FPGA is transmitted to the Slave FIFO. The Slave FIFO read-able and writable function is successfully simulated.

According to the analysis in Figure 2, the GIFII interface is connected to the FPGA FIFO through the internal Slave FIFO, and the port connection is normal. According to the analysis in Figure 7, after the reset signal is set to 1, the read-able-out and writable signal changes alternately. When the read-out flag bit is lowered, the read-able enable signal is lowered, indicating that data is transferred from the bus to the FPGA. When the writable flag bit is lowered and the empty signal in the FPGA is not empty, the write- enable signal is lowered, indicating that data is transferred from the FPGA to the bus. The function simulation of the top-level module is successfully verified [18].

5. Challenges and future prospects

This paper realizes USB 3.0 communication based on FPGA, but it is limited to read and write operation of USB chip and has not realized direct communication between FPGA and PC. In the future, related exploration will continue, focusing on integrating the functions of the USB chip into the FPGA, and realizing the goal of direct communication between the FPGA and the PC through the USB interface. In this way, the types of expanders of FPGA peripherals will be further expanded.

It is worth mentioning that with the continuous progress of the times, faster transmission speed USB chips have come out. In future studies, we will actively consider the adoption of newer USB chips to meet the increasing information transmission requirements. Through the updated USB chip, the USB communication interface will be able to better adapt to future development and provide faster and more stable communication services.

It is an important research direction to realize the direct communication between FPGA and PC. By integrating the functions of the USB chip directly into the FPGA, the advantages of the FPGA in peripheral control can be fully utilized. At present, FPGA has become an important hardware platform, widely used in various fields. The ability to communicate directly with the PC will further enhance the application range and flexibility of the FPGA. In the past, FPGA mainly communicated with PC through other interfaces, such as Ethernet and serial ports. Although such a communication method is feasible, it is limited by the characteristics of the interface and the transmission speed. Through the USB interface to communicate directly with the PC, you can make full use of the high-speed transmission characteristics of the USB standard, and further improve the efficiency and stability of communication.

In addition to the implementation of USB3.0 communication based on FPGA, we will continue to investigate other USB-related technologies and applications. For example, the rapid popularization and wide application of USB Type-C interface provides more possibilities for communication between FPGA and PC. We will explore how to use Type-C interfaces to achieve higher speed communication and improve the peripheral management and application capabilities of FPGA.

Future research will also aim to further expand the field of application of FPGA. In addition to USB communication, FPGA can also connect to a variety of other peripherals, such as displays, audio devices, and more. This will provide users with more diversified application scenarios and meet the needs of different fields.

6. Conclusion

In this scholarly contribution, it employs the CYUSB3014 chip, a product of Cypress Semiconductor, to architect a communication interface capable of actualizing USB 3.0 communication in conjunction with Field-Programmable Gate Arrays (FPGAs). This intricate interface establishes a conduit between personal computers and host systems, facilitating the transfer of data. Notably, data relayed from the host system to the FPGA is systematically conserved within the FPGA's First-In-First-Out (FIFO) memory buffers.

The cogency and accuracy of the proposed communication interface are substantiated in a twofold manner: firstly, through the schematic representation delineated by the specialized software, and secondly, by the simulated temporal sequence diagrams. This methodological approach ensures a preliminary yet robust verification of the interface’s operational logic and functional correctness.

Significantly, the communication interface is engineered to harness the full potential of FPGA’s parallel data transmission capabilities, synergistically combined with the high-velocity transfer rates intrinsic to USB 3.0 protocols. By doing so, it presents a sophisticated solution for high-speed data transmission challenges, thereby marking a notable advancement in this domain. Concurrently, this innovation is poised to broaden the horizons for peripheral augmentation of FPGA systems, catalyzing a plethora of new opportunities for enhancements and expansions in the realm of digital system design and communications. The implications of this development are far-reaching, potentially ushering in a new era of efficiency and integration in high-speed data transfer methodologies.

Authors Contribution

All the authors contributed equally, and their names were listed in alphabetical order.

References

[1] Sung G M, Tung L F, Wang H K, et al. USB transceiver with a serial interface engine and FIFO queue for efficient FPGA-to-FPGA communication. IEEE Access, 2020, 8: 69788-69799.

[2] Tian F, Li J, Sun X, et al. Design and Implementation of USB3.0 Data Transmission System based on FPGA. Wuhan Zhicheng Times Cultural Development Co., Ltd. Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application

[3] Chenxi Zhou, Guoqiang Zeng. Design of high-speed data transmission interface based on USB 3.0. Computer measurement and control, 2020, 28(05): 146-150.

[4] Najmul A B M, Mohammad K, Tashfia A. FPGA Implementation of USB 3.0 (Super Speed Bus) Function IP Protocol using Verilog HDL. Planetary Scientific Research Center Conf. 2013.

[5] Lan Zhang. Design of high-speed video image acquisition and processing system based on FPGA and USB3.0. Hefei University of Technology,2018.

[6] Han Wang. Hardware design of USB3.0 high-speed real-time data acquisition and recording system. University of Electronic Science and Technology of China,2016.

[7] Shaobo Yang, Dongxing Pei, Xiaozhong Yue. Design of USB3.0 data transmission interface in high-speed data acquisition system. Electron device, 2015, 38(04): 912-916.

[8] Ning Z, Sun Y. Design of an FPGA-based USB 3.0 device controller. arXiv preprint arXiv:2301.11505, 2023.

[9] Muxin Wang, Yezhao Lu, Lifan Mneg. Design of ultra-high speed data transmission system based on FPGA and USB 3.0. Modern electronic technology, 2022, 45(16): 71-74.

[10] Kaisheng Fu. Research and design of high-speed data acquisition system based on FPGA and USB 3.0. Nanjing: Nanjing University of Aeronautics and Astronautics, 2018.

[11] Guosong Chen. Design and implementation of image information acquisition and processing system based on FPGA and USB 3.0. Changchun: Jilin University, 2019.

[12] Semiconductor C. PSoC® 5LP Development Kit Guide. San Jose, 2013b, 2011, 58.

[13] Choi B Y. Implementation of FPGA Verification System with Slave FIFO Interface and FX3 USB 3 Bridge Chip. Journal of the Korea Institute of Information and Communication Engineering, 2021, 25(2): 259-266.

[14] Yang Cuicui, Zhu Xiangdong, Li Fan. PC based on the start agreement with the FPGA design of communication system. Journal of electronic science and technology, 2014, 27(10): 136-138.

[15] Liu Linxian, Qiao Nannan, Tong Qiang et al. USB3.0 Communication Interface Design based on FPGA [J]. Journal of Test Technology, 2019, 35(03): 261-265.

[16] Wu Chunchun, Hu Huaixiang, Jin Da et al. Design and implementation of USB3.0 Communication Architecture based on FPGA. Computer and Modernization, 2017(10): 121-126.

[17] Kuang Peng, Liu Chong, Wang Yonggang. The common data transmission system based on FPGA and start design. Micro computer and application, 2017, 4 (7): 26-28+34.

[18] Wang Xinmu, Lu Yezhao, Meng Lifan et al. Design of ultra-high speed data transmission system based on FPGA and USB 3.0. The modern electronic technology, 2022, (16): 71-74.