A Design and Implementation of Custom Communication Protocol Based on Aurora

BING LI
School of Integrated Circuits
Southeast University
Sipailou No.2, Nanjing, Jiangsu Province, China
Southeast University Chengxian College
Dongda Road No.6, Nanjing, Jiangsu Province, China
CHINA

JIAJIN ZHANG，SHUILING YAN，WEI SHAO
School of Integrated Circuits
Southeast University
Sipailou No.2, Nanjing, Jiangsu Province, China
CHINA

Abstract: - This paper presents the design and implementation of a new custom communication protocol based on Aurora combining PCI Express bus. This design can be applied to the transmission and forward of data among multiple nodes in the network. This custom protocol uses serial connection and transfer data in the form of packet. This custom protocol is divided into four layers, respectively the transaction layer, network layer, data link layer and Aurora layer. The retransmission and flow control mechanism are introduced to guarantee the correctness and completeness of the data transmission, and the network layer supports the interconnection of multiple nodes and rapid forwarding of data. This design is verified successfully on the Modelsim simulation platform and completes the FPGA board level test. The device mode is XC7VX485T and the package is ffg1761. The serial transmission rate can achieve 10Gbps on the FPGA board.

Keyword: - Aurora protocol; PCI Express; custom protocol; Packet; Node; FPGA

1 Introduction

The development of Internet along with the advent of the big data results in the increasing demands for information interaction between devices in the network. How to realize the resource sharing and ensure its reliability, completeness, high-speed is becoming a hot research now [1-3]. Aurora protocol is scalable throughput, low resource cost and high speed serial point to point communication protocol [4]. It provides a transparent axis physical layer interface to users. It is quicker, less delay relative to PCIE, but its function is not perfect especially on data integrity [5]. The link layer of PCI Express can effectively guarantee the correctness and completeness of the data transmission and also the flow control mechanism can prevent the data loss of sudden data peak. The routing

mechanism based on node ID can make fast data forwarding between many endpoints come true. This paper introduces a custom communication protocol based on Aurora protocol combining the retransmission and flow control mechanism of PCI Express and adding the IP layer [6-9].

2 Principle of Custom IO

This protocol uses serial connection and transfer data in the form of packet. This protocol is divided into four layers, respectively the transaction layer, network layer, data link layer and Aurora layer. The data packets are sent out from the sending node through transaction layer, network layer, data link layer and Aurora layer. And then the packets pass through the Aurora layer, data link layer and network layer of the routing node. And according to the selected transmission port in the network layer of routing node, the packets are transmitted to the destination node. The data transfer process is shown in Fig.1.

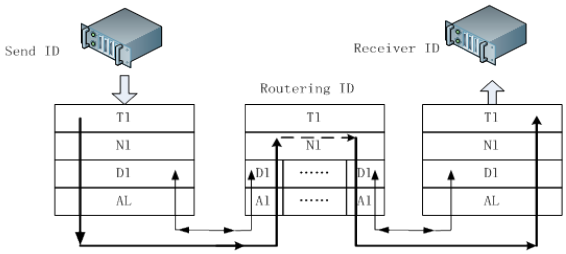

Fig.1 Transmission diagram

2.1 structure analysis

The main function of the transaction layer is to receive the encapsulated TLP packets from the application equipment and send to the IP layer. A TLP packet consists of packet header and payload. TLP header contains some information such as transfer type, routing information, transfer address and so on. The TLP packet format is presented in Fig.2.

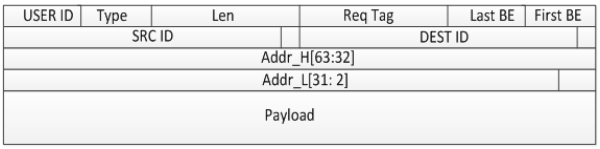

Fig.2 TLP packet format

USER ID is defined as the identification of the user. Type is defined as the type of request; SRC ID is defined as the ID number of source node which initiates a request. DEST ID is defined as the ID number of destination node which responds the initiated request. Req Tag is defined as a memory read packet identification, in order to receive CPLD packet correctly. Len is defined as the size of the packet. Addr is defined as the access memory for memory read or memory write. Last BE decides the effective bytes of the first double word and First BE decides the effective bytes of the last word. The length of payload should not exceed 1024 bit.

The network layer solves the problem of route transfer. It receives the packet from the TLP layer or from the data link layer and extract the destination node ID and then check the router table entries for the path to the destination node. The whole custom communication protocol structure is shown in Fig.3. The network layer connects to several data link layers. Routing lookup is for the best link and completes the data forwarding.

The main function of the data link layer is to guarantee the reliability of the data transfer. The data packet from the IP layer will be added the Sequence Number and CRC and encapsulated into a link layer data frame. The ACK/NAK agreement ensures the reliable transmission of a message and the flow control mechanism based on credit ensures to make full use of network bandwidth. Besides, DLLP (Data Link Layer Packet) is defined to transmit the ACK/NAK information and flow control

information.

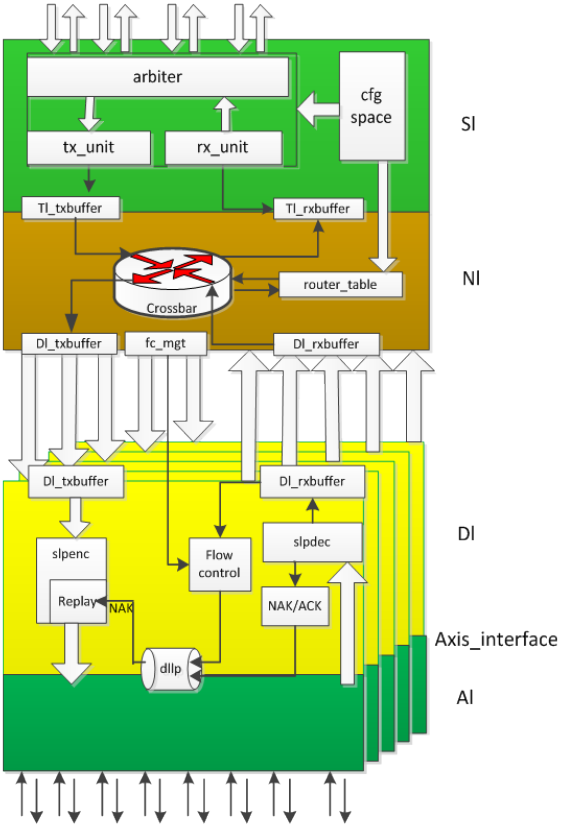

Fig.3 Structure of custom communication protocol

The Aurora layer provides media for data communication and provides reliable physical environment for data transmission. In this paper, a Xilinx IP core is used as the Aurora layer. Aurora protocol provides a simple transparent user interface for users. There are two kinds of model, flow model and frame model and the design in this paper is frame structure. It supports industry standard protocol.

3 Implementation

3.1 Transaction Layer

The main function of the transaction layer is to receive the encapsulated TLP packet from the application and send the packets to network layer. Also the transaction layer can receive the packet from network Layer and send to the application. The transaction is layer mainly composed of four components. Sending unit is responsible for sending the packets to network layer. Receiving unit mainly works for receiving the packets from network layer. Arbiter unit is in charge of arbitrating for packets from several devices, and choose the corresponding channel for data transfer by using round-robin algorithm. Configuration unit make some configurations of control registers and status registers and determines the operating parameters.

3.2 Network Layer

The main function of network layer is to set up the social connects and relationships among nodes. First of all, network layer reads the TLP packet which is stored in transfer buffer from transaction layer or data link layer, and extracts the information of destination node from the packet header. Then it will look up route table to decide which port the packet to be transmitted and write packet into the selected transfer buffer. Network layer consists of two major parts: routing module and crossbar control module. The core component of the routing module is routing table. The routing table is contained in the routing control module of network layer and totally controlled by external device. The routing algorithm is not set in network layer, which can realize the separation of data plane and network equipment, control network traffic flexibly and extend conveniently. Network layer can be used for data forwarding and storage stably as a simple and general bottom layer. The upper layer can control the transfer path of packet through updating routing table flexibly. As shown in Fig.4, the routing method is based on node ID and PORT ID represents the corresponding node ID number. When transaction layer sends a

request, network layer analyzes the request from local transaction layer or one of several data link layers, then looks up routing table to decide which port the packet to be transmitted. Finally, it realizes the fast data forwarding.

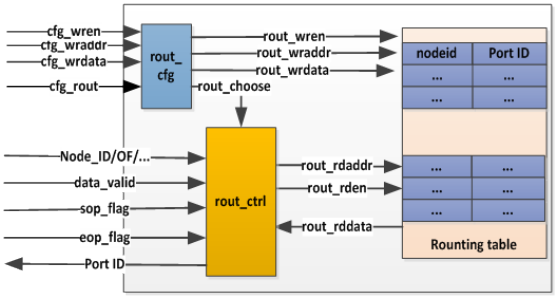

Fig.4 Routing Mechanism

The routing module translates the routing information from the local transaction layer or one of the several data link layers into the corresponding transfer request and then the request is input to crossbar control module. The crossbar control module makes two judgments to determine whether to respond to the request. One is to judge whether the link layer or the transaction layer which is corresponding to the current request is busy, and the other is to judge whether the corresponding data transmission path buffer has enough margin. If any one of the two needs has not been met, the initiated request will not be responded and the request needs to be issued again next cycle. If the above two conditions are both satisfied, the crossbar module will make the next judgment, checking if there are multiple requests to the same link layer or the local transaction layer. If not, only one request can be responded. The crossbar module will output the gating signal to respond the request and inform the corresponding data link or transaction layer. If there are multiple requests to the same link layer or the local transaction layer, the crossbar will cope with multi requests on the basis of round-robin arbitration and the other requests which have not been responded need to launch requests again next cycle, as shown in Fig.5.

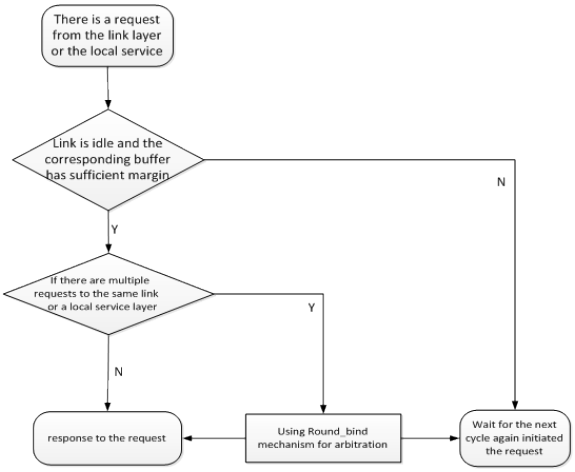

Fig.5 flow diagram of crossbar module

3.3 Data Link Layer

The data link layer is designed mainly to guarantee a reliable transmission of TLP packet. In the data link layer, ACK/NAK mechanism is to ensure the integrity of transmission on the both side of the link; CRC checking mechanism is to ensure the correctness of data transmission; flow control mechanism is to ensure that data does not overflow the buffer on the receiving side. When the packet is sent out from the data link layer, sequence number will be added at the start position of packet and CRC checking operator is added at the end position.

This section introduces the basic principle of ACK/NAK protocol. The receiver will check the sequence number and LCRC when it receives a TLP packet. According to the test results, the DLLP packet is feedback to the sender. After the sender receives the DLLP packet, according to the feedback information, the sender decides whether to retransmit the packets from the reply buffer.

The basic principle of flow control mechanism is illustrated in this section. According to the capacity of receiving

buffer in the data link layer, the receiver calculates credit values of each TLP packet and feedback to the sender through DLLP packet. The send status is controlled by the updated credits value and credit values of packets which are ready to be sent, in order to prevent the receiving buffer overflow. The flow control module is designed in data link layer in order to prevent the receiving buffer overflow and ensure the mass quantity transmission, convenient to multilink communications. The detailed implementation is shown in Fig.6.

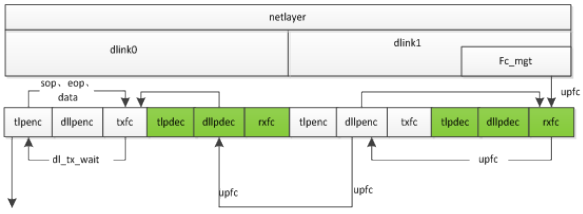

Fig.6 flow control mechanism

The slpenc module is responsible for encapsulation and sending of TLP packet, the slpdec module is responsible for receiving and parsing of packet, the dllpenc is responsible for sending of DLLP packet, the dllpdec is responsible for receiving of DLLP packet, the txfc module and rxfc module complete the process of flow control, and the fc_mgt module in network layer is feedback to the data link layer whether the packet is received correctly. The rxfc module analyzes the feedback signal which is collected from network layer by the fc_mgt module and the credit value can be got. After that, the updated credit value is transmitted to the other end by the dllpenc module. In the other end, the dllpenc module parses the correct credit value and transmits to the txfc module. After that, the txfc module gets the credit values of the packet which has been sent according to the tlpenc. The difference of the total received credits and the sent credits decides whether the signal dl_tx_wait is valid. The signal dl_tx_wait is an input signal of module tlpenc and determines directly whether the data continues to be sent, which prevent the overflow of the buffer in the receiving side of the other end and stop the transmission of data from sending buffer while the flow control is needed. The flow control is designed in data link layer, and combining with network layer, can control the flow control problem of multi links effectively.

3.4 Aurora layer

Aurora 64B/66B is an extensible, lightweight, flow control optional link-layer protocol for high-speed serial communication. It also defines two kinds of the user interface, respectively flow model and frame model. The frame structure is shown in Fig.7.

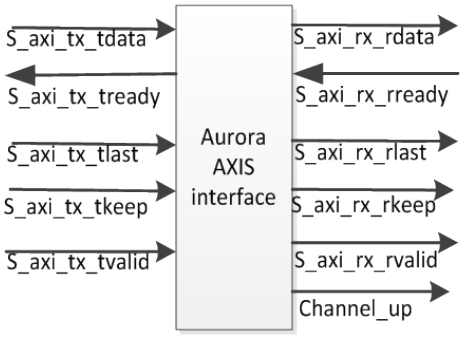

Fig.7 AXIS interface of Aurora layer

The data handshaking transmission is implemented through the signal ready and signal valid. The signal channel_up indicates the successful link establishment and inform the upper layer can perform the transmission and processing of data. The Aurora built-in flow control module has not been used, because the flow control mechanism has been designed in data link layer.

4 Experiment and Result

In Fig.8, a whole test model of the custom protocol stack is presented. Four Xilinx VC 707 board are connected to represent the

interconnection of four nodes. The software ChipScope in PC is used to grab the test waveform. The CORE represents the detailed structure of each node, which is shown in Fig.9.

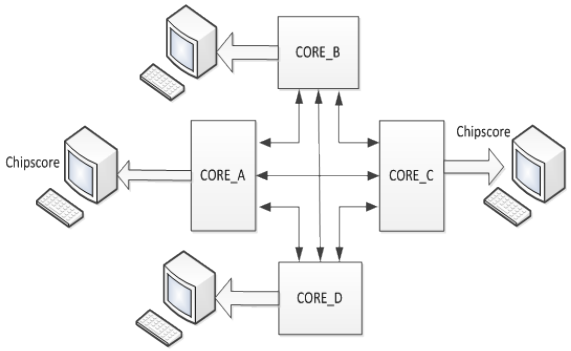

Fig.8 test model

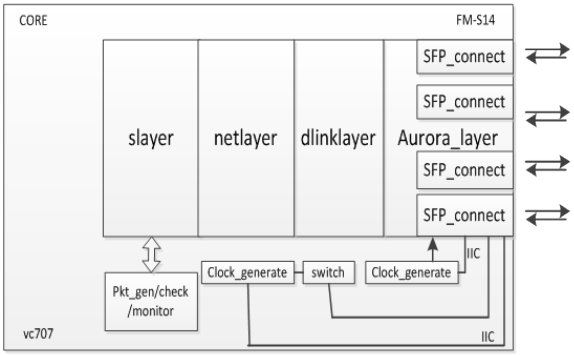

Fig.9 Structure of CORE

In Fig.9, the pkt_gen/check/monitor module connects to transaction layer directly. This module can generate packet and receive packet. The generated packet is sent orderly and reserved in each node, which is used to check whether the received data is correct. The clock module and reset module are linked to Aurora layer directly. The frequency of reference clock is consistent with the clock of FM-S14 sub-card switch, and the clock of other layers is relative to the serial rate. The user clock is output clock of MMCM (Mixed-signal clock management) unit which is drived from the clock recovered by the CDR (Clock and Data Recovery) unit. Here the serial rate is 10Gps.The waveform grabbed by ChipScope is shown in figure 10.The signal tx_cnt represents the transmitted packet numbers of current node and the rx_cnt represents the received numbers. The signal mrc_err and the signal mr_err_cnt represent the compared result of packet. Signal tl_tx_sop0, signal tl_tx_eop0 and signal tl_tx_data0 represent the sent data and corresponding control signal in the transaction layer. Signal tl_rx_sop0, signal tl_rx_eop0 and signal tl_rx_data0 represent the received data and corresponding control signal in the transaction layer. Signal LANE_UP and signal CHANNEL_UP represent the working state of Aurora. And when they are valid, it means the Aurora link training is completed and works normally. The signals related to AXIS represent the AXIS interface provided by AURORA IP, which conform the Aurora specification. As is shown in figure 10, the whole projects can complete the high speed data transmission correctly and successfully. The resource utilization is listed in Table 1.

Table 1 Resource utilization

Name	Used Number	Total Number	Ratio
Register	19731	607200	3%
LUTs	34253	303600	11%
LUT-FF	14666	39318	3%
IOBs	27	700	3%
BUFG/ BUFGCTRL/ BUFHCEs	6	200	3%
RAM/FIFO	15	1030	1%

Table 2 shows the relationship between transmitted packet numbers and time. When the working time is between 0 to 10us, the link is not built, so there is no packet sent. When the time is between 10us to 20us, the buffer space is larger enough, so it can send a large amount of packet continuously, ns. When the time is over 20us, the flow control mechanism works and the packet is sent stably. Figure 11 shows the line graph between packet numbers and time.

Table 2 Relationship between packet numbers and time

T(us)	Packet(number)
0	0
10	0
12	15
14	120
16	214
20	362
30	544
40	736
50	918
60	1093
70	1267
80	1445
90	1619
100	1791

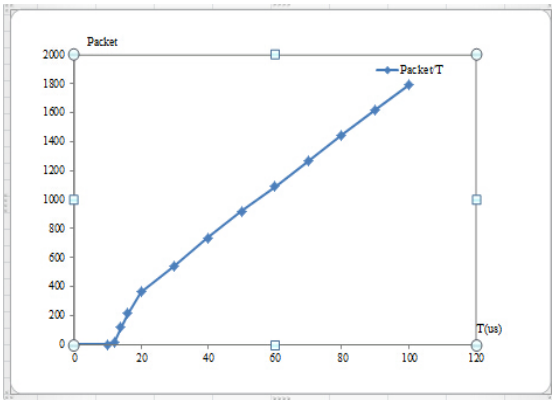

Fig.11 Line graph between packet numbers and time

5 Conclusion

This paper presents the design and implementation of a new custom communication protocol based on Aurora combining PCI Express bus in detail. The design can be applied to the transmission and forward of data among multiple nodes in the network. To achieve the routing mechanism based on node ID, the embedded network layer is introduced. Also the combination of network layer and data link layer can complete the data flow control. The entire implementation is finished with Verilog HDL. This design is verified successfully on the Modelsim simulation platform and completes the FPGA board level test.

Through the logic synthesis, the number of Slice LUTs is 34253, and takes up 11% of all, which means that the usage of resources is within a reasonable range. Through the board test, the serial rate can be up to 10Gbps. According to the statistics of the sending packets, the transmission rate of data packets are from fast to slow and ultimately to the stability in the whole transmission process. As it has been proven, the design solves the problems of resource sharing and finishes the function of quick and stable data transmission among multi nodes, and meets the design requirements.

Reference

[1] BrianTierney, EzraKissel, MartinSwany, EricPouyoul. Efficient Data Transfer Protocols for Big Data, 2012

[2] Hyoung Woo Park, Il Yeon Yeo, Jongsuk Ruth Lee. Study on big data center traffic management based on the seperation of large-scale data stream, 2013

[3] Weiqiang Sun, Fengqin Li, Wei Guo, Yaohui Jin and Weisheng Hu. Store, Schedule and Switch - A New Data Delivery Model in the Big Data Era,2013

[4] Zhou Dexiang, Zhang Liping. Study of Aurora IP Nuclear Communication Module Based on FPGA. 2011

[5] PCI Express Base Specification 3.0, November 10, 2010

[6] Diego Barrientos, Vicente González. Multiple register Synchronization with a High-Speed Serial Link Using the Aurora Protocol. IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL.60, NO.5, OCTOBER

2013

[7] Diego Barrientos, Vicente Gonzalez. Multiple Register Synchronization with a High-Speed Serial Link Using the Aurora Protocol, IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL.60, NO.5, OCTOBER 2013

[8] Zhongqi Li，Amer Qouneh. Aurara: A cross-layer Solution for Thermally Resilient Photonic Network-on-Chip IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, December 23, 2013

[9] Edin Kadric ,Naraig Manjikian,Zeljko Zilic. An fpga implementation for a high-speed optical link with a pcie interface

[10] Yi-HuangHung,Hung-YiLi,Po-YangHsu Yi-YuLiu. Dangling-wire Avoidance Routing for Crossbar Switch Structured ASIC Design Style,2010

APPENDIX_A

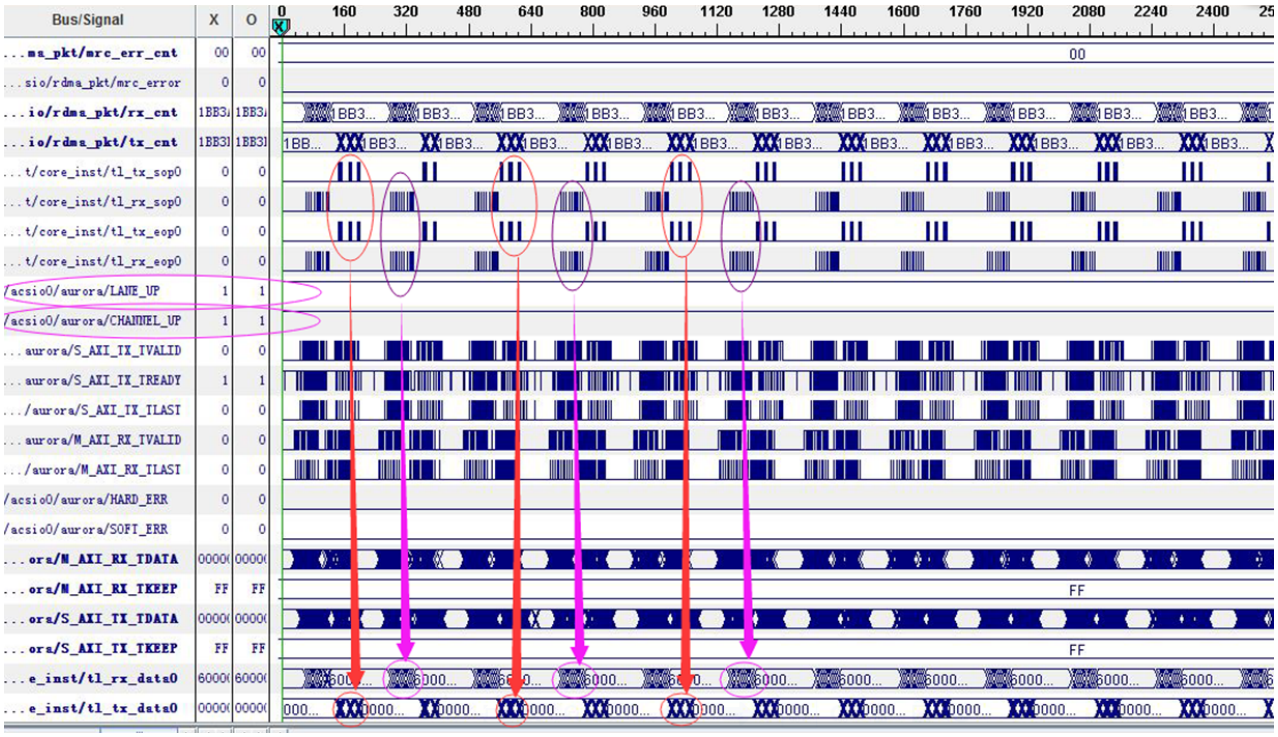

Fig.10 Waveform grabbed by ChipScope