Implementation of Aurora Interface using SFP+ Transceiver

Sungkyun Shin, Soyeon Choi, Eunchae Lee, Songyeon Lee, and Hoyoung Yoo
Dept. of Electronics Engineering
Chungnam National University
Daejeon, Republic of Korea
sgshin.cas@gmail.com, sychoi.cas@gmail.com, eclee.cas@gmail.com. sylee.cas@gmail.com, hyyoo@cnu.ac.kr

Abstract— **Aurora is a lightweight serial communication protocol for multi-gigabit links that allows data to be transferred between devices using one or more Giga-bit transceivers. Aurora protocol is implemented in duplex or simplex, and Aurora TX lane or Aurora RX lane must be implemented in two devices, respectively. Data transmitted from the user application in the TX lane go through link-layer frame delineation, 64B/66B encoding, and serialization, and data received in the RX lane are transmitted to the user application through deserialization and 64B/66B decoding. In this paper, the Aurora protocol between Xilinx FPGAs is implemented using two Kintex-UltraScale+ evaluation boards. As a result, it is proved that high-speed data transmission and reception between different FPGAs is possible through the Giga-bit transceivers.**

Aurora interface; Xilinx FPGA; Giga-bit transceiver

I. Introduction

Aurora is a lightweight serial communication protocol for multi-gigabit links [1]. One or more Giga-bit transceivers (GTX/GTH/GTY) can be used to transfer data between devices. The Aurora protocol refers to the transmission of user data through an Aurora channel composed of one or more Aurora lanes as shown in Fig. 1, and each Aurora lane can constitute a duplex or simplex serial data connection [2]. Aurora can be used in a variety of applications due to its low resource cost, scalable throughput, and flexible data interface. Aurora applications include chip-to-chip links between chips with high-speed serial connections, and board-to-board links that enable connections between devices using encoding schemes compatible with many existing hardware standards and backplane links[2]. Aurora protocol transmits 66-bit data with a 2-bit sync header added through the encoding process when 64-bit data is entered from the user application. The sync header includes information on whether the corresponding block is a data block or a control block. A 64-bit data block consists of 8 frames, where the unit of frame is composed of 8-bit [2]. The transmission rate can be determined by the circuit designer within the range of 0.5-28.01664 Gbps. In this paper, the Aurora protocol is implemented on two Xilinx evaluation boards to verify data transmission and reception between two boards.

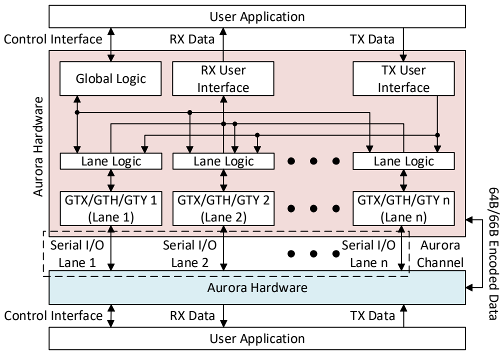

Figure 1. Hardware configuration of Aurora protocol.

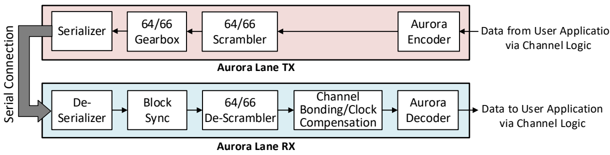

Figure 2. Aurora Lane for TX and RX.

II. Aurora Protocol

A. Hardware Configuraiton

Aurora protocol is implemented in duplex or simplex, and Aurora TX lane or Aurora RX lane respectively must be implemented in two devices. Figure 2 shows the simplex aurora channel implemented as a single lane, and the operation inside the TX lane and the RX lane together. Aurora lane TX consists of Aurora encoder, 64/66 scrambler, 64/66 gearbox, and serializer. Encoder creates a 66-bit data block by adding a 2-bit sync header to the 64-bit data block. The 64/66 scrambler performs self-scrambling based on the algorithm of IEEE 802.3ae [3] only for 64-bit data blocks except for the sync header. The 64/66 gearbox combines the 64-bit output from the scrambler with the sync header again. The serializer converts the order of bits to the reverse of standard [3] according to the bit-byte ordering convention. Aurora RX lane consists of de-serializer, block sync, 64/66 de-scrambler, channel bonding/clock compensation, and Aurora decoder. Since Aurora RX lane is received through a serial connection, deserialization is performed to convert serialized data back to parallelized data.

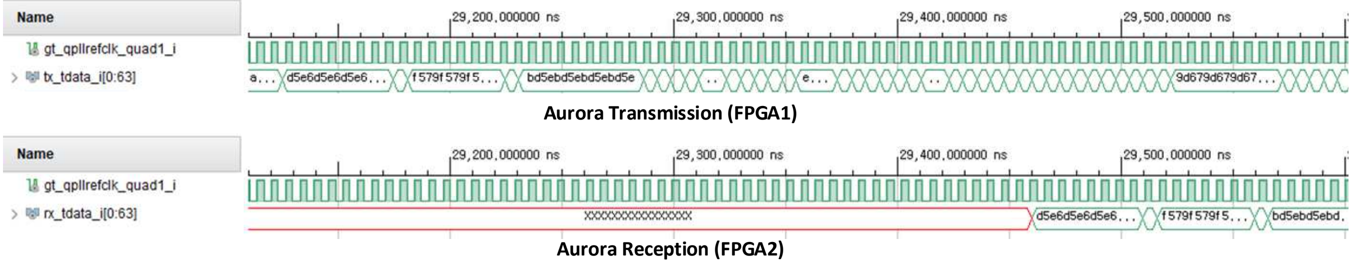

Figure 3. Operation of Aurora transmission and reception.

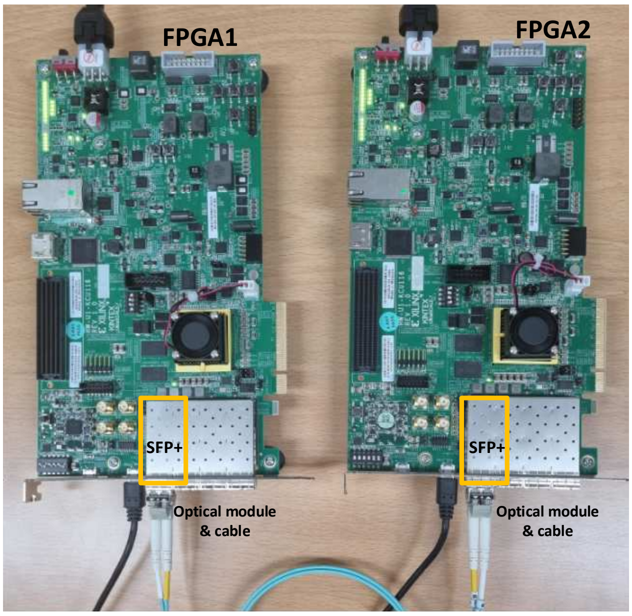

Figure 4. Connection of two evaluation boards.

Physical connection of Aurora lane can be implemented using SMA connector or SFP/SFP+ transceiver on the FPGA evaluation board. The SMA connector is connected with a 50Ω SMA cable, and the SFP/SFP+ transceiver is connected using optical modules and optical cables.

B. Data transmission and reception

In order to transmit and receive data through the aurora protocol, initializing process for the aurora channel is required. The initialization process consists of lane initialization and channel bonding. When all lanes are reset in the lane initialization process, it is ready to the channel bonding process. Channel bonding compensates the skew between lanes. When channel bonding is completed, in the case of simplex, it can freely transmit or receive data. In the case of duplex, it waits for the other party to prepare, then transmits or receives data.

Data received from the user application can be expressed in frames and blocks. A frame is a unit composed of 8 bits, and a block is composed of 8 frames. The data block is composed of frames through the link-layer frame delineation process, and the separator block is composed of bytes. Each data block is encoded including a sync header through a 64B/66B encoding process. The encoded data needs to be serialized. Since the data reception process reverses the transmission process, deserialization is performed. Then, the deserialized data block is received through a control block stripping process that classifies the type of control blocks and separates a sync header and a data block through 64B/66B decoding.

III. Experimental Results and Concolusions

Experiments are conducted using Xilinx FPGA to implement Aurora interface. Data transmission/reception between two KCU116 boards with built-in Kintex UltraScale+ chip is implemented. For circuit implementation, Vivado Design Suite 2021.2 is used, and Aurora hardware[3] is used. Aurora hardware is set to use one lane, lane rate is set to 10.3125 Gbps. The Giga-bit transceiver reference clock is set to 156.25MHz, and data flow mode was set to simplex. Figure 3 is configuration of hardware using two KCU 116 boards and SFP+ transceiver with optical modules and cable. Figure 4 shows the data transmitted and received through Aurora protocol in two FPGA evaluation boards. As shwon in Figure 4, when 64’hD5E6_ D5E6_D5E6_D5E6 is transmitted from TX of Aurora protocol implemented in FPGA 1, the data is received succesffuly from RX of FPGA 2.

In this paper, it was confirmed that complete data was transmitted and received by implementing the Aurora interface that supports communication between FPGAs. Therefore, by using the Aurora interface, it is possible to implement high-speed data transmission and reception using the Giga-bit transceiver of FPGA.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A5A8026986).

References

[1] C. Mani Pradhitha, and S. Kolangiammal, "Development and Implementation of Parallel to Serial Data Transmitter using Aurora Protocol for High Speed Serial Data Transmission on Virtex-7 FPGA," Indian Journal of Science and Technology, 2018.

[2] Xilinx, "Aurora 64B/666B Protocol Speficication," 2014.

[3] IEEE, "IEEE Standard for Information technology," in IEEE Std 802.3ae-2002, vol., no., pp.1-544, 26 Aug. 2002.