

# Design and Implementation Of High Speed Serial Data Transmission Techniques Using FPGA: RocketIO And Giga-Bit Ethernet

Parvathy V R<sup>1</sup>, Rejani Krishna P<sup>2</sup>, Merin Sarah George<sup>3</sup>

<sup>1</sup>Mar Athanasius College Of Engineering, Kothamangalam

<sup>2</sup>Scientist D, NPOL, DRDO, Cochin

<sup>3</sup>Asst. Professor, ECE Department Mar Athanasius College Of Engineering, Kothamangalam

Abstract—This paper proposes design and implementation of two high speed serial data transmission techniques such as RocketIO and Giga-bit Ethernet using architectural features of Virtex-5 FPGA. The data transmission and reception in RocketIO uses Aurora protocol on Multi-Gigabit Transceivers (MGTs). A 16-bit data from a Virtex-5 evaluation board is transmitted to other board using aurora protocol. Multi-Gigabit Transceivers present in the FPGA are configured using aurora protocol and works at the clock rate of 125 MHz (MGT clock). Aurora protocol converts the parallel data into serial data and vice versa. The data from aurora module is transmitted serially at the rate of 3.125 MHz through a SATA cable which is used as transmission medium for high speed serial data transmission. The receiver module receives data serially, and passed to a FIFO module to convert the 16-bit data to two 8-bits which can be given as a Giga-bit ethernet MAC input. This received data is transmitted to a PC using Giga-bit Ethernet and the received data is viewed in Wireshark.

## I. INTRODUCTION

The parallel transmission technology was widely used previously. In this technology data is transmitted in the form of parallel data, using multiplexer. In this format parallel lines are used for transmitting and receiving data which would consume more resources and time when compared to serial transmission.

Serial transmission technology is increasingly used for the transmission of digital data due to its improved signal integrity and high speed. Serial data transmission is suitable for communication between two participants or several participants. High-Speed Serial transmission [1] gained its importance because parallel I/O schemes will meet physical limitations when data rates begin to exceed just 1 Gb/s and can no longer provide a reliable, cost effective means for keeping signals synchronized. Serial I/O-based designs offer many advantages over parallel implementations which includes fewer device pins, reduced board space requirements, fewer printed circuit board (PCB) layers, smaller connectors, easier layout of the PCB, lower electromagnetic interference, and better noise immunity.

Multi-Gigabit Transceiver (MGT) is a Serialiser

/Deserialiser(SerDes) capable of operating at serial bit rates above 1 Gb/s. MGTs are used increasingly for data communications because they can run over longer distances, using fewer wires, and thus have lower costs than parallel interfaces with equivalent data throughput. MGTs are hard silicon which is present inside the FPGA. MGTs use new technologies to operate at high line rates besides serialization and deserialization, such as differential signalling, MOS current mode logic (MCML), emphasis, phase locked loops (PLLs), error detection, channel bonding and Electrical Idle/Out-of-Band Signalling.

## II. AURORA PROTOCOL

Aurora is an area-efficient, scalable data transfer protocol for high speed serial links. There are two types of aurora protocol depending on the type of encoding and decoding. They are Aurora 8b/10b protocol Aurora 64b/66b protocol

The Aurora 8B/10B [2] protocol is used for the implementation of our application. It is a scalable, link-layer protocol that can be used to move data from point-to-point across one or more high-speed serial lanes (Lane is a full duplex physical serial connection). The Aurora 8B/10B is protocol independent and can be used to transport industry standard protocols, such as Ethernet and TCP/IP, or proprietary protocols. This allows designers of next generation communication and computing systems to achieve higher connectivity performance. The Aurora 8B/10B protocol is primarily targeted for chip-to-chip and board-to board applications and it can also be used for box-to-box applications with the addition of standard optical interface components. It is an open standard and is available for implementation by anyone without restriction.

The LogiCORE IP Aurora 8B/10B core [3] implements the Aurora 8B/10B protocol using the high-speed serial transceivers on the Virtex-5 LXT, SXT, FXT, and TXT family, the Virtex-6 LXT, SXT, CXT, HXT, and lower power family, and the Spartan-6 LXT family. The Aurora 8B/10B core is a lightweight, serial

communications protocol for multi-gigabit links. It is used to transfer data between devices using one or many GTP/GTX transceivers. Connections can be full-duplex or simplex.

Aurora 8B/10B cores automatically initialize a channel when they are connected to an Aurora channel partner. After initialization, applications can pass data freely across the channel as frames or streams of data. Aurora frames can be any size, and can be interrupted at any time. Gaps between valid data bytes are automatically filled with idle sequences to maintain lock and prevent excessive electromagnetic interference. Flow control is optional in Aurora and can be used to reduce the rate of incoming data, or to send brief high priority messages through the channel.

Streams are implemented in the Aurora 8B/10B core as a single, unending frame. Whenever data is not being transmitted, idle sequences are transmitted to keep the link alive. The Aurora 8B/10B core detects single-bit, and most multibit errors using 8B/10B coding rules. Excessive bit errors, disconnections, or equipment failures cause the core to reset and attempt to re-initialize a new channel. The Aurora 8B/10B protocol describes the transfer of user data across an Aurora 8B/10B channel. An Aurora 8B/10B channel consists of one or more Aurora 8B/10B lanes. Each Aurora 8B/10B lane is a full-duplex serial data connection. The devices that communicate across the channel are called channel partners.

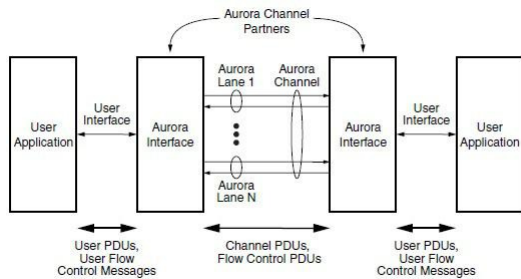


Fig. 1. Aurora Channel Overview

In the figure 1, it shows that how the communication between two user applications through aurora cores. It also shows that the aurora lanes and channels where the serial transmission of the data is transmitted between the users. The following are the uses of the aurora protocol: 1.High bandwidth transmission limited only by SERDES data rate capability. 2.Supports a large number of bonded lanes for high aggregate bandwidth. 3.Supports Full Duplex and Simplex channels. 4.Unlimited Frame size/flexible framing. 5.Aurora IP Core minimizes FPGA resource utilization.

III. APPLICATION DESIGN

The following figure explains the design and its procedure to implement the high speed data transmission serially.

The resources used for design and implementation of the high speed data transmission are as follows: A. Software

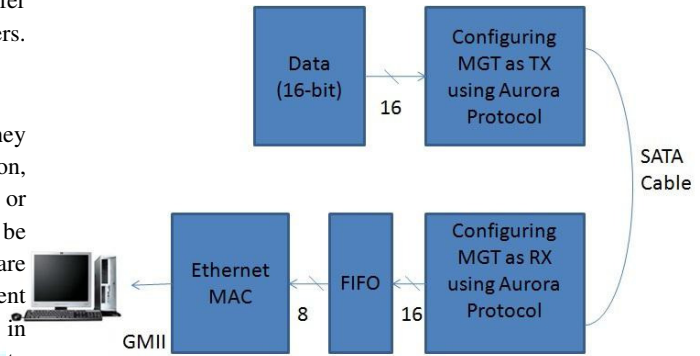


Fig. 2. Block Diagram

resources: 1. Xilinx 13.1 ISE tool. 2. Xilinx core generator. 3. Wireshark B. Hardware Resources: 1. Virtex-5 RocketIO development boards. 2. SATA Cable 3. JTAG cable 4. Ethernet Cable The Xilinx FPGA device used is virtex-5 LXT series (XCV5VLX50T-1FF1136) with speed grade -1 and package ff1136. It is used for the logic implementation of high speed serial connectivity.

The different platforms in which virtex5 devices present are LX, LXT, SXT, TXT and FXT platforms. Each one is specified for particular purpose depending on the application.

1. Virtex-5 LX: High-performance general logic applications
2. Virtex-5 LXT: High-performance logic with advanced serial connectivity
3. Virtex-5 SXT: High-performance signal processing applications with advanced serial connectivity
4. Virtex-5 TXT: High-performance systems with double density advanced serial connectivity
5. Virtex-5 FXT: High-performance embedded systems with advanced serial connectivity
6. Gigabit transceivers are present inside the FPGA silicon.
7. GTP and GTX are the two types of transceivers present in the virtex 5. RocketIO GTP transceivers [4] are present in LXT and SXT Platforms which supports speeds in the range of 100 Mb/s to 3.75 Gbps baud rates. RocketIO GTX transceivers are present in TXT and FXT Platforms which supports speeds in the range of 150 Mb/s to 6.5 Gbps baud rates. Each transceiver supports full-duplex, clock and data recovery in virtex-5 series.

IV. IMPLEMENTATION

The application is implemented in the following Phases 1. AURORA Protocol Implementation and modification phase. 2. Ethernet Implementation and modification phase.

A. AURORA Protocol Implementation and modification phase

Multi-gigabit transceivers [3] are present in FPGA as hard silicon. These are configured using aurora protocol for the application of high speed serial data transmission at the rate of 3.125 GHz. One of

the important parameters that we should select while generating the core is LaneWidth which decides the number of bytes transmitted during the period of one clock cycle. Aurora supports LaneWidth of 2 and 4. If we set the LaneWidth to 4, the user clock (USER CLK) should be half of the reference clock (REFCLK), whereas with a LaneWidth of 2, the frequency of USER CLK and REFCLK should be the same. So, the effective data rate will be same for both cases. Moreover, the clock generation for a LaneWidth of 4 is complicated than for a LaneWidth of 2. So we have selected the default LaneWidth value of 2. There are two types of data path interfaces used for the core which are Framing and Streaming. The Streaming interface is a simple word-based interface with a data valid signal to stream data through the Aurora channel whereas the Framing interface is a Local Link interface that allows encapsulation of data frames of any length. Since in our application we have to transmit data continuously over the communication link, we selected framing interface. The Aurora core supports two data flow modes such as Simplex and Duplex. We have selected Duplex mode because we need to transmit data in both directions. The Aurora core also supports several clock inputs to drive the RocketIO transceivers. REFCLK is low jitter differential clock that can support line rates up to 3.125 Gb/s. It is selected to match the transceivers clock. Now, select the MGT tile whose connections are brought outside to connect SFP module case for data transmission.

The program was modified separately for transmission and reception as the test design does it in a loop back mode.

#### B. Ethernet Implementation and modification phase

Ethernet interfacing is implemented in VHDL making use of the Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC wrapper which automates the generation of HDL wrapper files for the Embedded Tri-Mode Ethernet MAC (Ethernet MAC) in Virtex-5 LXT, SXT, FXT and TXT FPGAs using the Xilinx CORE Generator software. Although, VHDL and Verilog instantiation templates are available in the Libraries Guide for the Virtex-5 FPGA Ethernet MAC primitive; however, due to the complexity and large number of ports, the CORE Generator software is preferred as it simplifies integration of the Ethernet MAC by providing HDL examples based on user-selectable configurations. The program is modified separately for transmission and reception.

The Ethernet MAC wrapper file [5] instantiates the full Ethernet MAC primitive. All unused input ports on the primitive are tied to the appropriate logic level and all unused output ports are left unconnected.

The Ethernet MAC wrapper file instantiates the full Ethernet MAC primitive. All unused input ports on the primitive are tied to the appropriate logic level and all unused output ports are left unconnected. The Ethernet MAC attributes are set based on the options selected in the CORE Generator. Only used ports are connected to the ports of the wrapper file. This simplified wrapper is used as the instantiation template for the Ethernet MAC in the design. The design includes an Address

Swapping Module and a FIFO. Frames received by the Ethernet MAC are passed through the Receive side of the FIFO. Data from the Receive side of the FIFO is passed into the Address Swap Module and then on to the Transmit side of the FIFO using a LocalLink interface. The Transmit FIFO queues frames for transmission and connects directly to the client side Transmit interface of the Ethernet MAC. The address swap module is modified so as to send the data received from the FIFO. Then the data is sent to PC, which has Wireshark software to capture the data packets.

#### V. CHIPSCOPE PRO ANALYSER

ChipScope Pro[6] is software-based logic analyser. It allows to monitor the status of the selected signals in a design in order to detect possible design errors. It provides several cores that can be added to a design by generating the cores with the CORE Generator tool, instantiating them into the source code, and connecting the cores to the design before the synthesis process. Alternatively, it is possible to customize the cores and insert them into the design netlist using the ChipScope Pro Core Inserter tool after the synthesis process. The design is then implemented into the FPGA device using the implementation tools of the Xilinx.

The type of cores required for using ChipScope Pro analyser are 1. ICON (Integrated CONTroller) 2. ILA (Integrated Logic Analyser) and 3. VIO (Virtual Input/Output) ChipScope Pro Cores use the JTAG Boundary Scan port to communicate with the host computer through a JTAG downloading cable (either a parallel or a USB cable).

#### VI. RESULTS

The following are the results obtained for the integrated AURORA module in chipscope analyser. In the figures 3 and 4, the signals shown are transmitted data and received data. To keep the channel link continuously active, idle sequences are sent when actual data is not present. The channel up and lane up signals, which are active during transmission, were given to LEDs. So when connection was established, LEDs lighted up showing the successful transmission and reception.

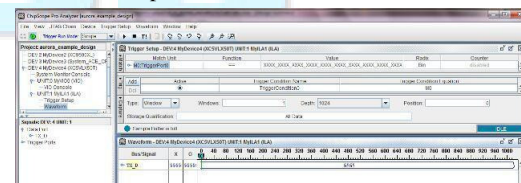


Fig. 3. Transmission of data viewed in Chipscope Analyzer

The received data which is given to Giga-bit ethernet is further sent to PC. This packets which are viewed in Wireshark are shown in figure 5.

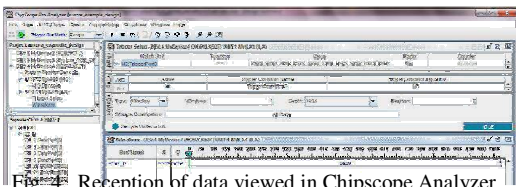


Fig. 4. Reception of data viewed in Chipscope Analyzer

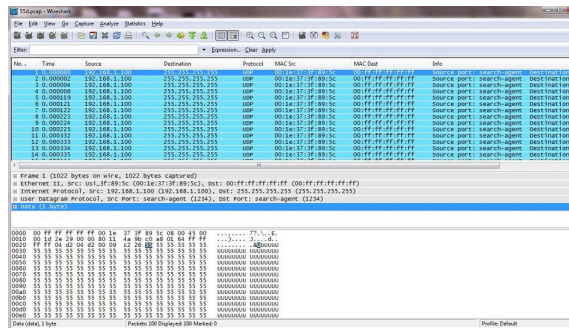


Fig. 5. Received data packets viewed in Wireshark

## VII. CONCLUSION

The high speed serial data is transmitted through SATA cable at the rate of 3.125 Gbps using multi-gigabit transceiver and received by other multi-gigabit transceivers. Both the transmitted and received data are monitored in the chipscope analyser. This received data was sent to PC using Giga-bit Ethernet and was verified by viewing in wireshark.

## REFERENCES

- [1] High-Speed Serial I/O Made Simple, A Designers Guide with FPGA Applications by Abhijit Athavale and Carl Christensen, Xilinx, 2005.
- [2] Aurora Protocol Specification SP002 (v2.0) September 19, 2007.
- [3] LogiCORE IP Aurora 8B/10B v5.2 User Guide, UG353 (v5.2 ) July 23, 2010.
- [4] Virtex-5 FPGA RocketIO GTP Transceiver User Guide, UG196 (v1.8.1) December 1, 2008.
- [5] Virtex 5 FPGA Embedded Tri-Mode Ethernet MAC User Guide, UG194, October 1, 2009.
- [6] ChipScope Integrated Bit Error Ratio Test (IBERT) for Virtex-5 GTP, DS773 April 19, 2010.