# Design and Implementation of TSEMAC and UDP/IP Network Stack on FPGA

**Bhavika A. Vithalapara, Abhimanyu Dhiman, Sudhir Agrawal, Shailendrasinh Parmar**

***Abstract*— This paper presents a high speed FPGA implementation of Triple Speed Ethernet MAC (TSEMAC) and UDP/IP stack which is the stack of the widely used in transport streaming and video conferencing applications. The Physical layer and its interface to the FPGA's IO blocks are implemented off-the–shelf using an integrated Ethernet transreceiver (National DP83848C). The link layer is based on Altera Triple Speed Ethernet MAC core. A novel architecture of network and transport layers by means of fabric and dedicated FPGA blocks is also proposed which can provides a PC-FPGA and vice-versa data communication. The design is analyze and tested using software program running in PC which send and receive data. The proposed system shows a noticeable speed up suitable for FPGA based data streaming applications.**

**Index Terms—** ***TSEMAC, UDP/IP stack, FPGA, PER Test Tool.***

## I. INTRODUCTION

Ethernet is synonymous with networking and its application is ubiquitous worldwide. It provides high bandwidth over long cable length and a driver-less architecture within an operating system environment [1]. It is a very popular and commonly used standard when connecting to a Internet or LAN(Local Area Network). In the past when connecting an system to a LAN, it was nessesary to use additional network circuits that had more functionality than required which came at higher cost [2]. Furthermore, to implement the network stack a processor was needed. Now, with the FPGA technology it is easy to implement an application-tailored part of a UDP/IP stack to achieve a cost-effective and straight-forword connection to a network.

The UDP/IP protocol has applications on audio and video streaming of VoIP, broadcasting, multimedia and video conference communications [3]. For real time conversations, UDP protocol provides low delay datagram transfers, due to unreliable service.

This work presents a TSEMAC and UDP/IP network stack implementation on FPGA and make a comparison with other exiting work. Here the packets are shown in wireshark to integrate with the application layer. This paper is organized as follows: Section II presents an overview of the OSI model layers. The proposed system and implementation is detailed on Section III, while evaluation and results are exposed in Section IV. Finally Section V conclude the whole work.

**Bhavika Vithalapara** B.E. E&C, pursuing M.E. degree in E&C taking internship at SAC, ISRO Ahmedabad for dissertation work of M.E. Her main research interest includes Networking, broadcast services and Wireless Communication.

**Abhimanyu Dhiman** B.Tech. E&C. Presently he is Scientist at Developmental Communication Technique Division, ISRO Ahmedabad.

**Sudhir Agrawal** B.E degree in E&C and MBA from Gujarat University, Gujarat, India.

**Shailendrasinh Parmar** B.E. in E&C, and M.E. from Rajiv Gandhi Technical University, Bhopal, M.P., India.

## II. OSI MODEL

The OSI(Open Systems Interconnection) model is a theoretical model and is used to describe the behavior of a network. the OSI model consists of seven layers and the layers are named: Application, Presentation, Session, Transport, Network, Datalink and Physical layer. From a TCP/UDP/IP viewpoint the session and presentation layers are often included in the Application layer.The OSI layers are frequently referred in this paper, but it's not further explained. For a detail description of the protocols and layers, see [4].

### *A. Data Link Layer*

The link layer provides the data exchange between computers within a local network. The basic unit of data transfer, for data link layer is data link packet frame. A frame is composed of a header, payload and trailer. It carries the souce address, destination address and other control information in the header. The trailer contains the checksum of transported data. By using the checksum, we can find out the error which has been occurred during transfer. The network layer packet is included in the payload.

### *B. Network Layer*

The network layer is responsible for host to host delivery of packet. It establish the route between originating and destination computer. The basic unit of data transfer is a datagram that is encapsulated in a frame. It is composed of a header and data field. This datagrams are the payload or data field of the frame. This layer is used to establish communication with computer systems that lie beyond the local LAN segment, because it has its own routing addressing architecture, which is separate and distinct from the link layer. Such protocols are known as routable protocols, for example IP (Internet Protocol) [5]. The IP is the important protocol of the Network layer.

### *C. Transport Layer*

The transport layer is responsible for communication between two applications running on different computers [6]. The basic transmission unit is a segment, that is composed of a header and payload, which is transmitted through payload of network packet. The transport layer provides end-to-end reliability by having flow and error control. It has two protocol: TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). TCP provides a connection-oriented communication with flow control, reliable data delivery and duplicate data suppression. Wheres UDP provides unreliable and connection less communication services. It allows applications to send datagram and handle

translation between sockets and ports. UDP is very effective for real-time applications like audio and video or in applications where low delay and low latency is preffered over reliable data delivery [7].

## III. The Proposed System and Implementation

The Hardware UDP/IP stack core is shown in Figure 1. Figure 1(a) shows the architecture of a traditional TCP/IP stack and Figure 1(b) shows the UDP/IP stack which are entirely implemented on FPGA.

| | |
|---|---|
| Application layer | Application layer |
| Transport layer | UDP |
| Network layer | IPv4 |
| Link layer | Ethernet |
| Physical layer | Physical Medium |

Figure 1(a) Architecture of TCP/IP stack, (b) used architecture

In our design, UDP protocol is implemented at transport layer. Internet Protocol Version 4(IPv4) is used at network layer, which gives more area effective design compared to IPv6 protocol.

The architecture of proposed design is shown in Figure 2. It consists of several modules. Starting with the physical layer, DP83848C PHY device can be used as an interface for Ethernet communication at 10/100 Mb/s speed. PHY connect to Ethernet cable through RJ-45 connector. MII interface is defined by IEEE 802.3 specification. When using a MII interface, PHY chip provides tx and rx clock to MAC. At MII interface that operate at 10 Mb/s, PHY provides 2.5 MHz clock and for 100 Mb/s speed it provides 25 MHz clock.

Triple Speed Ethernet MAC Megacore function is offered by Altera with high capability and in variety of operating modes. This function is configured to operate at half-duplex or full-duplex with minimum possible resource consumption and suitable for data streaming applications. PHY interface is managed by MDIO module. In our design we are using MII interface in full-duplex mode with 100 Mb/s speed. TSEMAC features and operation are illustrated in [8].

The Ethernet MAC function provides an Avalon-ST interface to user applications and an industry standard interface to external PHY devices. It can support upto 24 ports. We can configure single-port MAC to include internal FIFO buffers to store the data on transmit and receive path [8]. PHY provides a 25 MHz clock to TSEMAC. The PHY is managed by MDIO(Management Data Input/Output) module which uses the Management Data Clock (MDC) of 2.5MHz.

The Megacore function is constructed from two units, the transmitter and receiver unit. On the transmit path, data transfer are synchronous to the rising egde of tx_clk. The tx_en signal is asserted to indicate the start of a new frame and remains high until the last byte of frame is present on tx_d[3:0]. If an error occurred in the frame during transmission, it subsequently transmitted with the tx_err for one clock cycle.

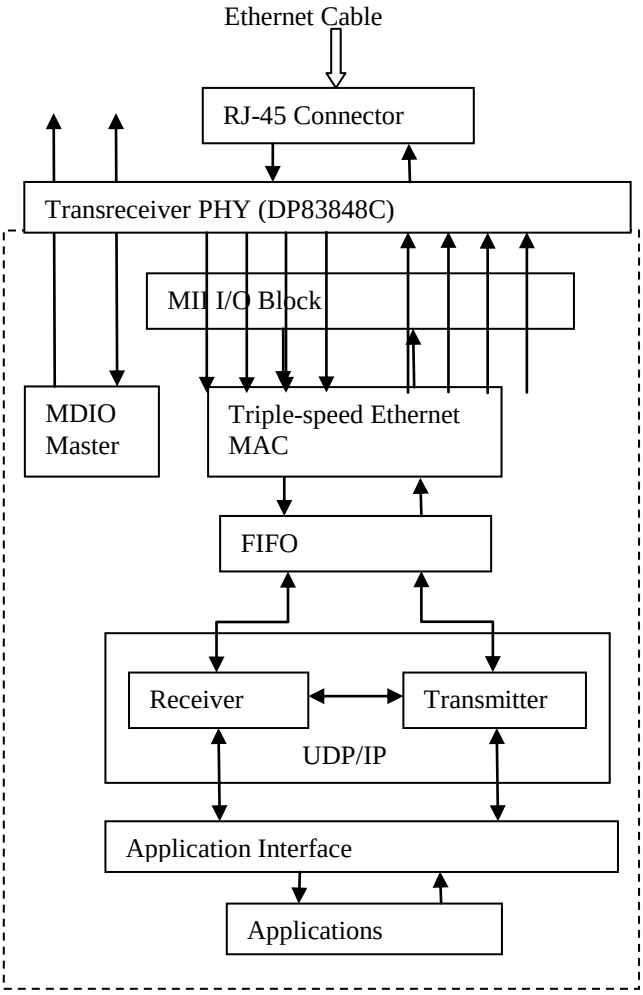


Figure 2 Proposed Design Architecture

On the receiver path, all signals are sampled on rising edge of rx_clk. The PHY assert the rx_en signal to indicate the start of a new frame and remains high until the last byte is present on rx_d[3:0]. If there is an error in the frame received from the line, the PHY assert the rx_err for one clock cycle.

## IV. Evaluation and Results

The Hardware system is designed in QSYS using Triple Speed Ethernet Megacore function. There are three systems: Ethernet main system, Ethernet subsystem and Peripheral system. Ethernet main system consists of NIOS II processor which control and configure all the modules, SDRAM which is use to store the data, flash memory to store the MAC address of device and peripheral and Ethernet interface systems. Ethernet subsystem consists of Ethernet bridge to connect this system with main system, TSEMAC to configure, control and access the PHY device with different speed and modes, SGDMA(Scatter Gathered DMA) tx/rx to send and receive data to and from FIFO of TSEMAC, and descriptor memory to store the data of DMA controller.
The peripheral system consisting of peripheral bridge to connect this system with main system, performance counter for debug and system performance analysis, system clock timer, and JTAG UART for serial communication and debugging NIOS II application via on-board USB-blaster

circuitry. Here the system ID block is used to sync the hardware system generation with software generation tools.
We uses one PLL(Phase Lock Loop) in our design which accept input clock of 50 MHz and generate 100 MHz system clock, 100 MHz SSRAM clock, 60 MHz peripheral clock etc. The software interface is provided by the Eclipse which is software tool for NIOS II processor. To transmit data SGDMA tx controller are open, assert tx_en signal and send data. Figure 3 shows the tx data view of signal tap analyzer which is real time analyser for system.

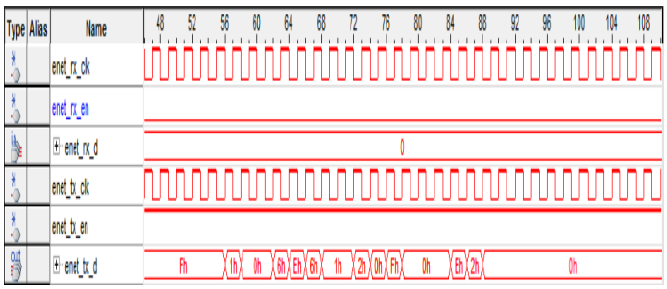


Figure 3 Tx data in signal tap analyzer

When frame is coming on receiver side PHY assert the rx_en signal and NIOS access this data through SGDMA rx controller. Figure 4 shows rx data in signal tap analyzer.

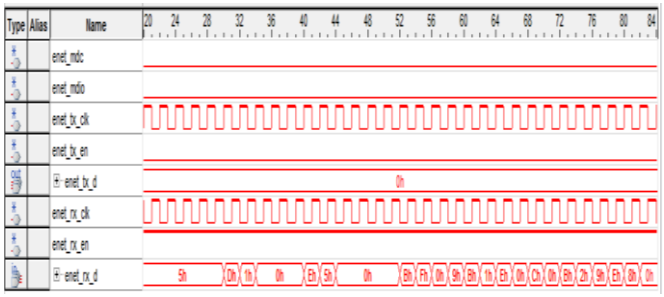


Figure 4 Rx data in signal tap analyzer

We implement nichestack which is a networking stack. It receive data from application and transform it into the network specific packets and send to the networking devices. Here the IP is assigned to the device which is 192.168.10.234 for our device and UDP port Number assigned is 30. Now when ping the device it gives the response as shown in figure 5 by ICMP packets.

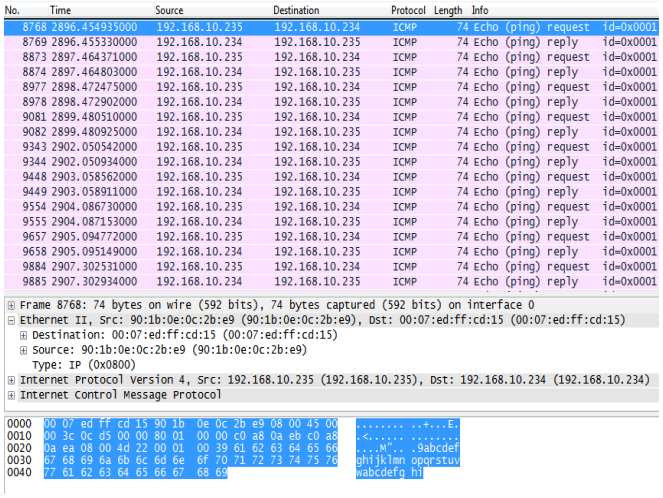


Figure 5 ICMP packets of ping request and reply

To check the UDP functionality we send the UDP packets to the device by PER (Packet Error Rate) test tools which measure the packet error rate as shown in figure 6, and set the local loopback on the device.

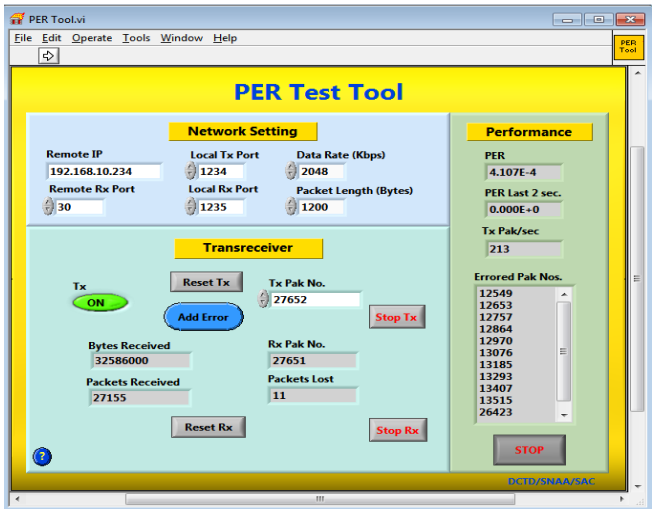


Figure 6 PER Test Tool

In the PER test tool we set the remote IP to FPGA device IP that is 192.168.10.234, and remote rx port to 30 because we set the listen port of the device to 30. Local rx and tx port are set to port no. 1234 and 1235 of host PC. Data rate is set to 2Mbps and packet length is 1200 byte. Now when Transreceiver is turn on it send and receive the packets from the device. Here it shows that among 27155 received packet 11 packets are lost. So packet error rate is very low. Figure 7 shows the UDP packets communication in wireshark.

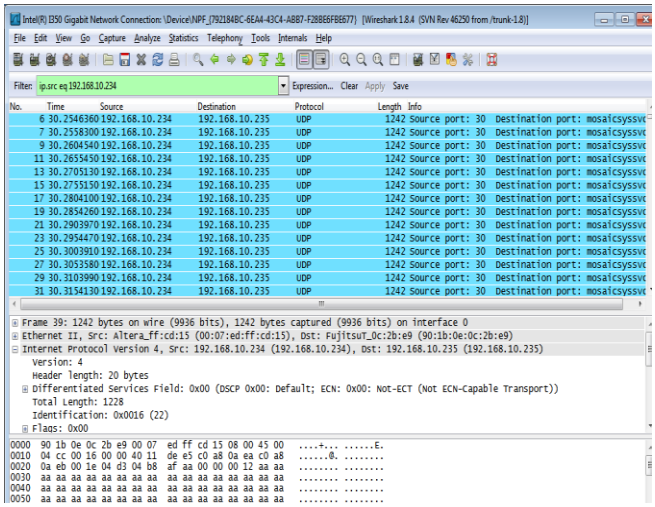


Figure 6 UDP data packets

We compare our work in terms of speed(MHz), Maximum Ethernet frame length (bytes) and maximum Ethernet speed (Mbps) with other implementations and we get the results that is summarized in TABLE I. They all implement the stack in Xilinx devices wheres we implement it in Altera device. In [2] they provide three different UDP/IP core: Minimum, Medium and Advanced. [9] presents complete TCP/IP stack implementation from which we observed the data only related to UDP/IP implementation. In [10], it only implement the Ethernet stack and [11] presents the complete UDP/IP core but there is neighter UDP checksum capability nor ARP protocol is considered.

TABLE I

| Ref.No | Speed(MHz) | Max. Frame Length (bytes) | Max. Ethernet Speed (Mbps) |
|---|---|---|---|
| Our | 125 | 1518 | 1000 |
| 2-Min. | 90.7 | 256 | 100 |
| 2-Med. | 60.3 | 256 | 100 |
| 2-Adv. | 105.6 | 1518 | 1000 |
| 9-Dollas | 77 | - | 100 |
| 10 | 50 | - | 100 |

In speed term our work gives the best performance. For maximum Ethernet frame length our implementation and advanced UDP/IP core are capable to manage the frame of 1518 bytes. [9] and [10] does not provide information about frame length. The maximum Ethernet speed is 1000Mb/s is supported by three designs as shown in table.

## V. Conclusion

This paper shows TSEMAC and Niche networking stack on FPGA. Here a UDP/IP stack is implemented and verified in Altera Cyclone III. It uses 78% of total logic element of FPGA and can operate at maximum frequency of 125 MHz. Design uses NIOS II processor to control and configure other modules. As resource efficiency is one of the main goal, a selected embedded processor with function as a CPU within the stack. The design is tested and analyzed in signal tap analyzer and wireshark. We implement the UDP/IP functionality using Niche stack where IP provides basic datagram delivery services, on top of which UDP delivers a multiplexing features so that multiple application can concurrently use UDP/IP stack. We also compare our design with other work in terms of speed, maximum frame length and maximum Ethernet speed. The results shows that our design gives higher speed among other designs. Other two parameters gives intermediate solution.

## Acknowledgments

Authors are thankful to Shri Kaushik Parikh, Dy Director, SNAA/SAC/ISRO and Shri Virender Kumar, GH, SSTG/SNAA/SAC/ISRO for his continuous guidance, encouragement and support. The authors sincerely appreciate the critical evaluation and constructive suggestions provided by the reviewers.

## References

[1] Marcus R Perrett and Izzat Darwazeh, "A Simple Ethernet Stack Implementation in VHDL to Enable FPGA Logic Reconfigurability", *International Conference on Reconfigurable Computing and FPGAs*, IEEE 2011.

[2] Andreas Löfgren, Lucas Lodesten, Stefan Sjöholm, Hans Hansson, "An analysis of FPGA based UDP/IP stack parallelism for embedded Ethernet connectivity", *In 23rd NORCHIP Conference*, pp. 94-97, IEEE 2005.

[3] Fernando Luis Herrmann, Guilherme Perin, Josue Paulo Jose de Freitas, Rafael Bertagnolli and Joao Baptista dos Santos Martins, "A Gigabit UDP/IP Network Stack in FPGA", *16th IEEE International Conference on Electronics Circuits and Systems (ICECS)*, pp. 836-839, IEEE 2009.

[4] A. S. Tanenbaum, *Computer Networks*, 4th ed. Prentice Hall, August 2002.

[5] K. S. Siyan and T. Parker, *TCP/IP Unleashed*, 3rd ed. Sams Publishing, August 2002.

[6] L. Dost´alek and A. Kabelov´a, *Understanding Tcp/ip: A Clear And Comprehensive Guide*. Packt Publishing, April 2006.

[7] P.-K. Lam and S. Liew, "UDP-liter: an improved UDP protocol for realtime multimedia applications over wireless links," in *1st International Symposium on Wireless Communication Systems, 2004.*, Sept. 2004, pp. 314–318.

[8] San Jose, "Triple Speed Ethernet MagaCore Function User Guide", *ALTERA, 101 Innovation Drive, Software Version 9.1*, November 2009.

[9] A. Dollas, I. Ermis, I. Koidis, I. Zisis, and C. Kachris, "An open TCP/IP core for reconfigurable logic," in *13th Annual IEEE Symposium on Field Programmable Custom Computing Machines, 2005. FCCM 2005.*, April 2005, pp. 297–298.

[10] Nima Moghaddami Khalilzad, Sheida Pourshakour, "FPGA implementation of Real-time Ethernet communication using RMII Interface", *3rd International conference on Communication Software and Networks (ICCSN)*, IEEE 2011.

N. Alachiotis, S. Berger, and A. Stamatakis, "Efficient PC-FPGA Communication over Gigabit Ethernet," Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on. IEEE, 2010.

**Bhavika Vithalapara** received the B.E. degree in electronics and communication from Gujarat Technological University, Gujarat, India, in 2013 and she is currently pursuing M.E. degree in electronics and communication from Gujarat Technological University, Gujarat, India. She is taking internship at SAC, ISRO Ahmedabad for dissertation work of M.E. Her main research interest includes Networking, broadcast services and Wireless Communication.

**Abhimanyu Dhiman** received the B.Tech. degree in Electronics and Communication from Deen Bandhu Choturam University of Science and Technology, Murthal, Haryana, India. Presently he is Scientist at Developmental Communication Technique Division, ISRO Ahmedabad. He is working on video streaming and broadcasting services for S-band satellite

**Sudhir Agrawal** received the B.E degree in electronics and communication and MBA from Gujarat University, Gujarat, India.
He has 20+ years of experience in satellite communication networking and protocols. Presently he is Head of Developmental Communication Technique Division at ISRO Ahmedabad. He is Dy. Project Director for Mobile Satellite Services (MSS) HUB and responsible for development of Reporting, Multimedia, Voice and Broadcast services for multi-beam S-band satellite. He is guide to graduate and postgraduate students from universities and has number of papers to his credit in national and international journals.
Sudhir Agrawal is Life Fellow of IETE (Institution of Electronics and Telecommunication Engineers), Member of ISTE (Indian Society for Technical Education) and ASCI (Administrative Staff College of India).

**Shailendrasinh Parmar** received the B.E. degree in Electronics and Communication from Bhavnagar University, Gujarat, India, and M.E. from Rajiv Gandhi Technical University, Bhopal, M.P., India.
He has 15+ years of experience in Academic field. Presently he is Head of Electronics and Communication department at Shantilal Shah Engineering Collage, Bhavnagar. His field of specialization are Fiber optics communication, Digital Signal Processing, Microprocessor and Comunication Engineering.