International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 1, January 2014

FPGA Based Platforms for Ethernet Data Transfer

Indu Raj, and Rejani Krishna. P.

Abstract: **This paper introduces a method of implementing Gigabit Ethernet data transmission and reception in FPGA, using the embedded processor MicroBlaze and compares it with the implementation based on the CORE Generator tool generated embedded Tri Mode Ethernet MACs. The system presented in this paper is implemented on the evaluation board ML505 which contain a Virtex-5 FPGA. Using the XPS tool, the embedded processor MicroBlaze is configured inside the Virtex-5 FPGA. The software part of the processor is configured using the lwIP Echo Server template available in SDK.**

***Index Terms*—Embedded software, Ethernet networks, Field programmable gate arrays, Microprocessors.**

I. INTRODUCTION

FIELD programmable gate arrays (FPGA) are flexible and reusable high density circuits. The important feature of these devices is the ability to reconfigure them. That is, any portion of the system can be reconfigured at any time while the rest of the design is still working. Xilinx provides a lot of tools which will help the designers to configure the FPGAs more quickly and easily.

The Virtex-5 FPGA has embedded Tri Mode Ethernet MACs. This can be accessed using the CORE Generator tool provided by Xilinx [1]. Thus, using this tool the MAC can be configured such that it can transmit and receive Gigabit Ethernet (GbE) packets.

With the advancement of FPGAs a new trend of implementing the microprocessors on the FPGAs has emerged in the design community [2]. The ML505 board supports the embedded processor MicroBlaze. By configuring the embedded processor MicroBlaze it is able to achieve the same transmission and reception of GbE packets that was achieved by configuring tri mode Ethernet MAC.

The paper describes how the GbE transmission and reception are achieved on the Virtex-5 FPGA using an embedded processor. The paper also compares the advantage of using the implementation based on embedded processor over the implementation based on the CORE Generator tool generated embedded tri mode Ethernet MACs.

Manuscript received October 18, 2013. This work was carried out at the Naval Physical and Oceanographic Laboratory (NPOL), Thrikkakara, Kochi, Kerala, India which is a national laboratory under the Defence Research and Development Organization (DRDO), Ministry of Defence, Government of India.

Indu Raj is Masters student in VLSI and Embedded System at Sree Narayan Gurukulam College of Engineering, Kadayiruppu, Kochi, Kerala , India

Rejani Krishna. P., Scientist, NPOL, Thrikkakara, Kochi, Kerala, India

II. EXISTING SYSTEM

The Tri Mode Ethernet MAC inside Virtex-5 FPGA when accessed using the CORE Generator tool generates example design. The behavior of the example design is specified using a set of HDL files. The design will receive Ethernet packets from outside. The Address Swap Module switches the destination address and source address within the received MAC frame and will transmit it back to the source address [3]. The Address Swap Module can be replaced with a User Logic block such that it is capable of receiving Ethernet data, modify it and transmit it to any other destinations required [4].

TABLE I
TRANSMIT FIFO LOCAL LINK INTERFACE SIGNALS

Signal	Direction	Clock Domain	
tx_ll_clock	Input	N/A	Read clock for local link interface
tx_ll_reset	Input	*tx_ll_clock*	Synchronous reset
tx_ll_data_in [7:0]	Input	*tx_ll_clock*	Write data to be sent to transmitter
tx_ll_sof_in_n	Input	*tx_ll_clock*	Start of frame indicator
tx_ll_eof_in_n	Input	*tx_ll_clock*	End of frame indicator
tx_ll_src_rdy_in_n	Output	*tx_ll_clock*	Source ready indicator
tx_ll_dst_rdy_in_n	Output	*tx_ll_clock*	Destination ready indicator

TABLE II
RECEIVE FIFO LOCAL LINK INTERFACE SIGNALS

Signal	Direction	Clock Domain	
rx_ll_clock	Input	N/A	Read clock for local link interface
rx_ll_reset	Input	*rx_ll_clock*	Synchronous reset
rx_ll_data_in [7:0]	Output	*rx_ll_clock*	Data read from FIFO
rx_ll_sof_in_n	Output	*rx_ll_clock*	Start of frame indicator
rx_ll_eof_in_n	Output	*rx_ll_clock*	End of frame indicator
rx_ll_src_rdy_in_n	Output	*rx_ll_clock*	Source ready indicator
rx_ll_dst_rdy_in_n	Input	*rx_ll_clock*	Destination ready indicator

For a transmission to occur correctly, the system must obey the signals shown in the Table 1. Successful transmission occurs when *tx_ll_src_rdy_in_n* and *tx_ll_dst_rdy_in_n* are asserted low. The individual packet boundaries are marked by

ISSN: 2278 – 909X

the *tx_ll_sof_in_n* and *tx_ll_eof_in_n* signals [3]. Similar to transmission, receiving system should also obey the signals shown in the Table 2. That is the reception occurs when *rx_ll_src_rdy_out_n* and *rx_ll_dst_rdy_out_n* are asserted low within the boundaries marked by the *rx_ll_sof_out_n* and *rx_ll_eof_out_n* signals. To simultaneously transmit and receive data, the system should obey all the signals that are required for both transmission and reception.

To realize the design using FPGA, using the Electronic Design Automation tool, a net list is generated from the HDL code. The net list is then fitted to the actual FPGA architecture using a process called place-and-route. Once the design and validation process is complete, the binary file generated is used to configure the FPGA. This file is transferred to the FPGA via a serial interface JTAG [5].

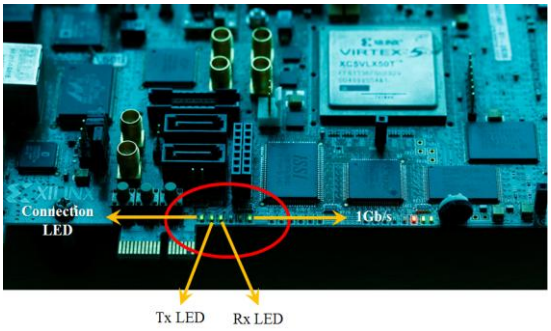

Fig. 1. Connection and Status LED behavior

After connecting the Ethernet cable from the PHY port of the evaluation board to the PC and transmitting the packets that is to be received on the board using the LabVIEW program, the Ethernet LED's in the ML505 board lights up indicating that Ethernet reception and transmission are in progress. The Ethernet connection LED, reception LED and the transmission LED light up along with the reception and transmission speed indicator LED, indicating 1 Gbps operation (see Fig. 1).

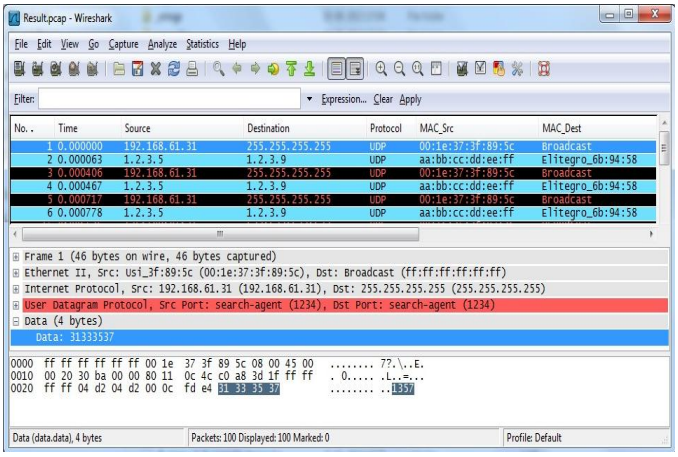

Fig. 2. Received frames captured in Wireshark

To check whether the reception and transmission is occurring correctly, the frames are captured using Wireshark. From the packets captured using Wireshark (see Fig. 2), it is noted that the packets that are broadcasted from the PC using the LabVIEW program are received on the board. The data that are transmitted are 1, 3, 5 and 7. By checking the data field of the packet, It was observed that the ASCII values of the data viz. 31, 33, 35 and 37, are received on the board. The board is programmed to convert this ASCII values into original data which is then transmitted as a 1 KB packet (see Fig. 3). The data is transmitted to the destination IP address 1.2.3.9. The IP address 1.2.3.5 is set as the IP address of our board.

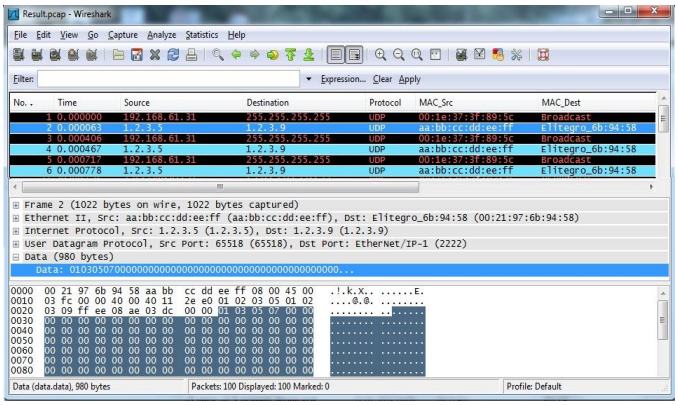

Fig. 3. Transmitted frames captured in Wireshark

III. PROPOSED SYSTEM

The implementation platform, ML505 evaluation board has Virtex-5 FPGA and it supports the MicroBlaze soft core processor. The system design is divided into two. One is hardware design, which includes the designing methods using Xilinx Platform Studio (XPS) and the other is Software design, which include designing methods using Software Development Kit (SDK) [6].

A. Hardware Design

Besides the MicroBlaze, other components required by the design are GPIO LEDs, a timer, an interrupt handling controller and a serial communication device UART. The MicroBlaze is implemented entirely in the general-purpose memory and logic fabric of FPGAs using the Embedded Development Kit (EDK) design environment. The BSB (Base System Builder) Wizard inside XPS is used for generating the embedded system around the MicroBlaze that is supported on the ML505 board. This wizard allows the selection of the board and processor. The initial parts generated were the MicroBlaze processor and its configuration. Here the system clock frequency is set to 125 MHz and the local memory to 64 KB. The next step was selection and configuration of peripherals, which include the DDR_SDRAM memory, the serial communication device RS232_UART, GPIO LEDs for display, Hard Ethernet MAC - xps_ll_temac, and timer xps_timer. The timer was used for the reference time generation part. Interrupt is enabled for both Hard Ethernet MAC and the timer. Interrupt handling was done with the help of the interrupt handling controller INTC named xps_intc_0. The instruction and data cache configuration is the final part.

B. Software Design

The software part of the design is configured using the SDK tool. For configuring the software portion the hardware design has to be exported to the SDK along with the BIT file. SDK

ISSN: 2278 – 909X

provides us with a great number of project templates that we can pick from. In this design the Ethernet MAC has to be configured inside the ML505 board such that it can receive UDP packets. So to meet the requirement, the lwIP Echo Server template is selected from the available templates [7].

The lwIP Echo Server template provides a simple demonstration on how to use the light weight IP stack. The server works on TCP data and listen for the input at the specified port and simply echoes back whatever data is sent to that port. Our requirement is to receive UDP packets, modify it and either transmit it or else display it on the LEDs. So the C-program is developed accordingly. In this design the IP address 192.168.1.10 is assigned to the board's MAC address, as the IP address of our PC is 192.168.1.100. Also the design is configured such that it will be listening for the input at the port 1234. Finally the FPGA board is connected to an Ethernet port on the host computer via an Ethernet cable.

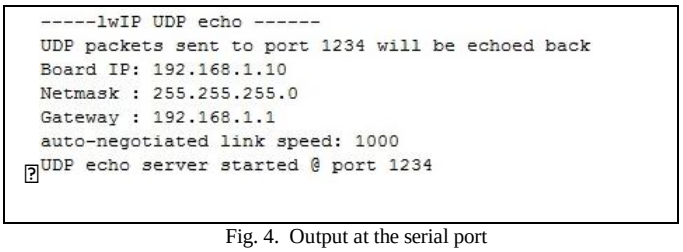

Fig. 4. Output at the serial port

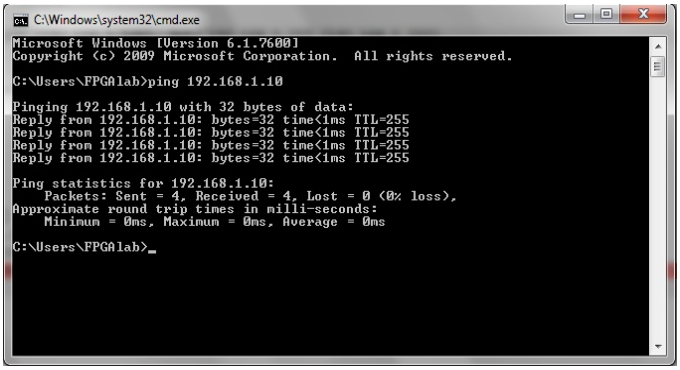

Fig. 5. PING result

The C code written in SDK is compiled with the GNU Compiler tool. The compiled C files along with the libraries generate the executable ELF (Executable and Linkable File) file. The final stage of designing is the association between the hardware and software parts and the download of the entire image into the FPGA. The Data2MEM tool links the BIT file generated at the end of hardware implementation and the compiled ELF file. The result is a download.bit file and this is downloaded into the FPGA. After successful download, the output in Fig. 4 can be seen at the serial port.

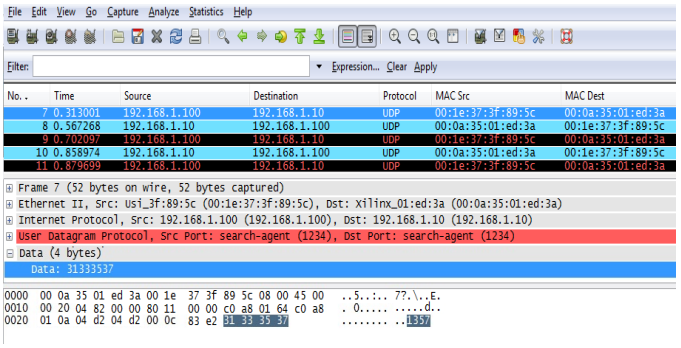

Fig. 6. Received frames captured in Wireshark

Now we will be able to ping to the IP address 192.168.1.10 from the PC. The ping result can be seen in the Fig. 5. There is a LabVIEW program which can transmit UDP packets to the IP address 192.168.1.10. So as the LabVIEW program is executed, it was observed that the Ethernet LED's in the ML505 board lights up, indicating that Ethernet reception and transmission are in progress. The Ethernet connection LED, transmission LED and reception LED lights up along with the reception and transmission speed indicator LED, indicating 1 Gbps operation.

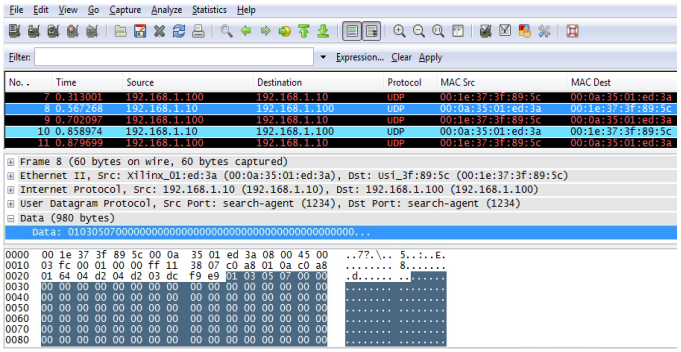

Fig. 7. Transmitted frames captured in Wireshark

From the packets captured using Wireshark (see Fig. 6), it is noted that the packets that are transmitted from the PC using the LabVIEW program are received on the board. The data that are transmitted are 1, 3, 5 and 7. By checking the data field of the received packet, it was observed that the ASCII values of the data are received on the board. Our design will convert the ASCII value to the corresponding decimal value and will transmit it back to the source address (see Fig. 7).

IV. Conclusion

The design based on both the CORE Generator tool generated embedded Tri Mode Ethernet MAC and embedded processor MicroBlaze are implemented successfully. The main advantage of the design using embedded processor is the reduced device utilization. From the device utilization summary available from both designs (see Fig. 8 and Fig. 9), it was observed that the design that used CORE Generator tool generated embedded Tri Mode Ethernet MAC, utilizes 43% of the device available while the MicroBlaze based design uses only 38%.

The motivation for the introduction of the MicoBlaze core comes from the idea that most FPGAs contained within an

ISSN: 2278 – 909X

embedded system require level of interaction with an external processor. Moving this processor onto the chip allows the FPGA and the processor to communicate without the bottlenecks associated with communicating with off-chip devices. This is also an advantage of this design.

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of LUT Flip Flop pairs used	943			
Number with an unused Flip Flop	240	943	25%	
Number with an unused LUT	289	943	30%	
Number of fully used LUT-FF pairs	414	943	43%	
Number of BlockRAM/FIFO	5	60	8%	
Number using BlockRAM only	5			
Number of 18k BlockRAM used	6			
Total Memory used (KB)	108	2,160	5%	

Fig. 8. Device utilization summary of implementation based on the CORE Generator tool generated embedded tri mode Ethernet MAC

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of LUT Flip Flop pairs used	12,036			
Number with an unused Flip Flop	3,390	12,036	28%	
Number with an unused LUT	3,969	12,036	32%	
Number of fully uscd LUT-FF pairs	4,677	12,036	38%	
Number of BlockRAM/FIFO	51	60	85%	
Number using BlockRAM only	51			
Number of 36k BlockRAM used	47			
Number of 18k BlockRAM used	6			
Total Memory used (KB)	1,800	2,160	83%	

Fig. 9. Device utilization summary of implementation based on embedded tri mode Ethernet MAC based on embedded processor MicroBlaze.

ACKNOWLEDGMENT

The authors wish to thank Director, NPOL for permitting to carry out this project. Authors also wish to thank Mr. Suresh M., Scientist G, NPOL and Mrs. Jayamma T. M., Scientist F, NPOL for their valuable guidance, help and insightful comments.

REFERENCES

[1] "www.fpgadeveloper.com" – Generating the Ethernet MAC.
[2] Xilinx Inc. MicroBlaze Reference Manual, UG081 (v13.2).
[3] Virtex – 5 FPGA GTP/GTX Transceiver User Guide – UG196.
[4] Indu Raj, Haripriya P., Rejani Krishna P.; Realization of Gigabit Ethernet Data Transfer in FPGA; Proceeding of the Third National Conference on Modern Trends in Electronic Communication and Signal Processing held at BPC College, Piravom, Ernakulam during 1-2 August 2013, p19-23.
[5] Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC Wrapper v1.4 – GSG340.
[6] Indu Raj, Haripriya P., Rejani Krishna P.; Data Transfer Using The Embedded Processor MicroBlaze; Proceeding of the International Conference on Emerging Trends in Engineering and Technology held at Munnar, Kerala during 4-5 October 2013.
[7] Xilinx, Embedded System Example, XAPP1026, version 2.2, 2006.

Indu Raj took her B.Tech. degree in Electronics and Communication Engineering from Cochin University of Science and Technology in 2009 and is presently in the final stage of her M.Tech. in VLSI and Embedded System.

Rejani Krishna P. took her M.Tech. degree from Cochin University of Science and Technology. She is currently working as Scientist in Naval Physical and Oceanographic Laboratory (NPOL), an underwater surveillance laboratory of the DRDO, Government of India.

ISSN: 2278 – 909X