IPASJ International Journal of Electronics & Communication (IIJEC)
Web Site: http://www.ipasj.org/IIJEC/IIJEC.htm
A Publisher for Research Motivation ..... Email:editoriijec@ipasj.org
Volume 7, Issue 4, April 2019 ISSN 2321-5984

Block Level Design Implementation of 100 Mbps Ethernet Telemetry using Vivado TEMAC IP core in Artix-7

Gowri J[1], Manoj G[2] and Karunakara P Menon[3]

[1]Gowri J, M.Tech in VLSI and Embedded Systems, RSET

[2]Manoj G, Sc. 'D', NPOL

[3]Karunakara P Menon, Assnt. Professor, RSET

ABSTRACT

Telemetry is used to acquire data from a remote location and transfer to a location where it is analyzed. In sonar systems, data acquired is transmitted and received at desired location using Ethernet. Here the Ethernet data coming from a location is received and it is transmitted to another location as needed. Designs in which IPs are added as RTL source files are a little complex for the modern IP which has complex interfaces and port mappings. Block level design of the receiver- transmitter system is implemented using Vivado IPs integrator tool and is implemented on ARTIX 7 FPGA evaluation board. Vivado IP integrator tool integrates the complex IPs in a single step.

Keywords: Ethernet, FPGA, UDP, Vivado, IP

1. INTRODUCTION

Telemetry systems are used to gather measurements at remote or inaccessible points and transmit them at receiving stations where those measurements can be monitored and analyzed. Telemetry has various applications: it is used in defense and space exploration, energy monitoring, agriculture, transportation, and various testing and measuring applications. Telemetry is most widely used in aeronautical applications where characteristics of a space shuttle, satellite, aircraft or UAV are monitored and transmitted to the ground stations. Sonar is a technique used in underwater vessels for navigation and communication. The data from the sensor is collected and transmitted to another location where it is processed. Ethernet is used as a communication bus as it supports extensive hardware (MAC controllers) and software (UDP stacks). Most commonly UDP protocol is used for data transmission. RTL based coding methods used earlier have complex port mappings, interfaces and signals which are a little difficult to implement. Also the earlier used platform ISE is no longer supported by Xilinx for newer families. Here we use block level design with Vivado IP integrator tool [1]. IP integrator instantiates all IPs and makes all the interconnections. The modern IPs consists of multiple complex interfaces and ports. Use of ISE design suite is complex and confusing. Using the IP integrator, the complex interfaces can be connected in a single step. The ports connections are made for each IP core and control signals are generated and added as a block in the design, in order to make the IP function.

2. ETHERNET TELEMETRY

Modern day FPGAs has a variety of options for communication, but the fastest and the most reliable is the Ethernet. Ethernet is the first local area network (LAN). It is a widely used network because of its speed, simplicity and ease of installation. Ethernet was used to interconnect computer workstations and to send data from computer to printers etc., Ethernet was published as a standard IEEE 802.3 Career Sense Multiple Access with Collision Detection. Ethernet MAC is present in FPGA either as hardware or as a soft IP core connecting to an external PHY. PHY is necessary to connect the Ethernet MAC to an external device.

Converting a data into Ethernet data can be done in two ways; either using microprocessors or by using Programmable Logic Devices (PLD) and the latter is Faster, reconfigurable and easy to design. FPGA Ethernet implements the MAC and PHY of the Ethernet protocol at 100 Mbps rate.

2.1. PROPOSED SYSTEM:

Proposed system is a block level design of Receiver Transmitter Ethernet telemetry system. It is designed using the IP integrator tool provided by Xilinx Vivado. The Ethernet data generated using LabVIEW is received by the TEMAC IP core and stored into a FIFO and BRAM. This data is again transmitted by the TEMAC by accessing the BRAM and is captured and viewed in the Wire Shark tool. The data for Ethernet telemetry used here is 100 Mbps.

2.2.1 FPGA AND TEMAC IP CORE:

Trimode Ethernet MAC IP core provided by Xilinx supports 10/100/1000 Mbps and 2.5 Gbps Ethernet MAC with half duplex and full duplex transmission. The IP core can be customized by selecting the necessary options according to our requirement. The TEMAC is selected for configuration vector mode with RGMII interface and full duplex operation. Here 100 Mbps Ethernet speed is used. Configuration vector can be set from the User guide for TEMAC IP core which is an 80 bit vector [3]. The configuration vector is set according to our LAN requirements and it enables the Ethernet transmission and reception. For Transmission and reception, the configuration vector varies. 125 MHz clock is used for the TEMAC IP core.

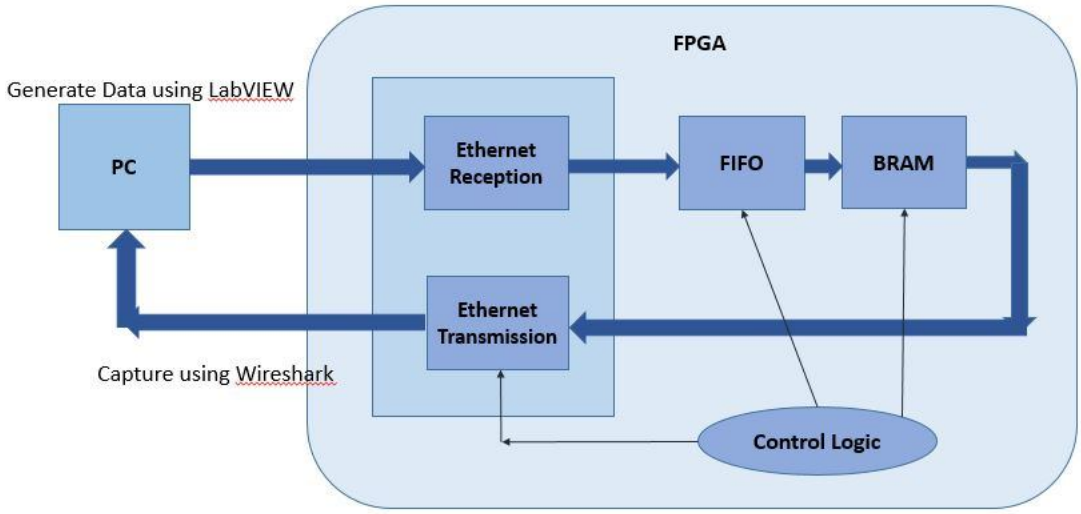

Figure 1 Block Diagram of the Proposed System

2.1.1 A) Ethernet Transmission:

Transmission using TEMAC requires the generation of control signals as given in the timing diagram. When data is to be transmitted the *tx_axis_mac_valid* is set to 1 and data is placed on pin *tx_axis_mac_data*. The first two bytes of data is accepted by the core by asserting *tx_axis_mac_ tready* and then the remaining frame is accepted. *tx_axis_tlast* is asserted on the final byte if the frame indicating the end of data. For 100 Mpbs RGMII interfaces the *tx_axis_aclk* is 125 MHz and *tx_axis_tready* is activated during every ten cycles. The controller for transmission is a state machine which implements the timing diagram functionality. The UDP header is included in the controller logic as an array, and transmitted just before transmitting the required data part which is read from a FIFO.

2.1.1 B) Ethernet Reception:

The IP must be ready to receive data at any time. The received data will be represented as the timing diagram shown in figure 3. The configuration vector for transmission can be written from the IP user guide and fed to the necessary pin for successful reception. The data to be received can be send from the PC using LabVIEW UDP sender. This generates a UDP packet and is transmitted which is received by the FPGA. In this receiver side we get signals like *rx_axis_mac_tvalid* and *rx_axis_mac_tlast* from the IP core along with the received data. The *rx_axis_mac_tvalid* indicates the valid ethernet data which is received and the *rx_axis_mac_tlast* represents the last byte of data which is coming out of the IP core. These can be observed in the timing diagram for reception. The timing diagram for Ethernet transmission and reception are as follows,

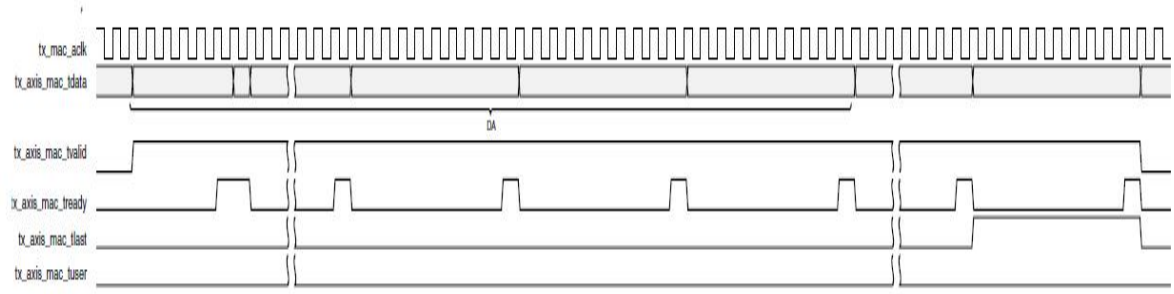

Figure 2 Timing Diagram for Normal Frame Ethernet Transmission

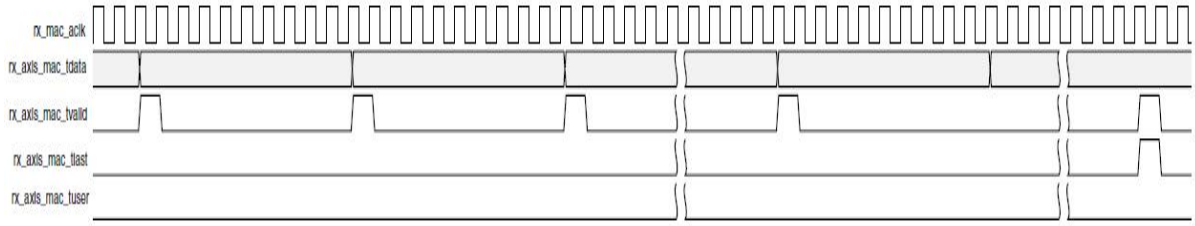

Figure 3 Timing Diagram for Normal Frame Reception

2.1.2 FIFO Generator:

FIFOs are memory buffers that are used to store data temporarily until another process is ready to read it. The first data that is written into a FIFO will be read out first. It is used when two processes operate in a different rate. An example is the high speed communication channel that writes into a FIFO and a slower channel that read data from FIFO at a slower rate. FIFO generator IP core presented by Xilinx can be configured to work for applications requiring storage and retrieval. The FIFO generator IP core can be customized to our required size. The control signals for FIFO can be generated accordingly for Write and Read operation. The FIFO also works in 125 MHz clock. The timing diagram for FIFO write and read operation [4] is used as reference to design the control block. Control logic for FIFO has signals like write enable, read enable, indices to indicate write and read pointers, full, empty and reset. Read and write operations can be implemented at different clock speeds by choosing independent clocking while configuring the IP. Writing data into FIFO occurs when the write enable is held high and reading when read enable is high.

2.1.3 Block Memory Generator:

The Block memory generator is used to create and optimize a Block memories for storing of data and retrieval in a system. FPGAs contain two columns of memory, called Block RAM. Block RAM is used for storing large amounts of data in the FPGA. Its is a dual port network consisting of read and write port and an address where the data is stored. Upto 18Kb of data can be stored in a Block RAM and also the length and width is customizable. It has control signals like clock, write enable, enable, and addresses into which the data is stored.

When the write enable is high, in each clock cycle, write data is written into an address location [5]. The written data can be read from the same location by making the write enable low. In Xilinx, the IP block memory generator functions in three modes, write first mode, read first mode and no change mode. Its is operated in 125 MHz frequency in no change mode.

2.1.3.a) Write first mode: Here the input data is simultaneously written into the memory and read out from the output port at the same time.

2.1.3.b) Read first mode: The data which is stored at the write address previously is obtained at the output, while data is stored into a memory.

2.1.3.c) No change mode: The data output is the previously read data, and is unaffected by the write operation.

2.1.4 Transmission control logic:

This control block is designed from the timing diagram which explains the working of ethernet transmission in TEMAC. The transmission controller (Figure 4) receives *tx_ready* from the TEMAC, *data_in* and *tx_en* from the reception controller. The *tx_en* signal gives indication that the reception and storing are finished and transmission can be started. Output signals are *re, data_out, tx_valid* and *tx_last. tx_valid* indicates the valid data for transmissionn and

tx_last shows the last data. *Data_out* is the data given to TEMAC for transmission, *re* is signal to read data from FIFO.

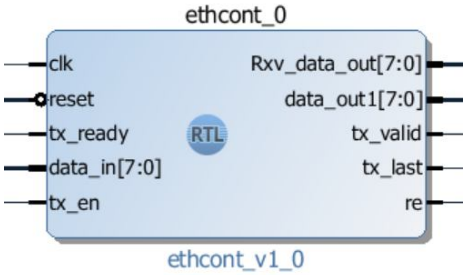

Figure 4 Transmission Controller Block For TEMAC

2.1.5 Reception control logic:

This block is designed according to the timing diagram logic of TEMAC reception. The receiver controller (Figure 5) receives the received data (*rx_data*), *tlast* and *tvalid* from the IP core. The *tlast* and *tvalid* indicates the valid data and last data of the received data respectively. en_out is the signal which enables transmission and *wr_en* signals writes the received data to the FIFO. *fifo_rst* is the reset generated for the FIFO.

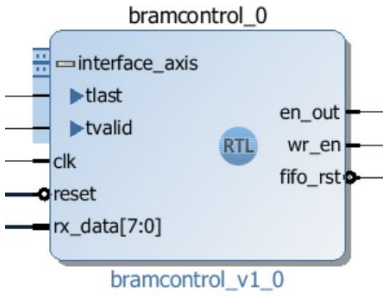

Figure 5 Control Logic Block For Reception And Storing Ethernet Data

2.1.6 Clocking Wizard:

The clocking wizard is the IP core which is used to produce the required output clocks of desired frequency, phase and duty cycle for the design. By customizing the IP core multiple output clock frequencies can be obtained for each IP used in the design with reduced jitter. Different set of output clocks are generated for different IP cores and the IP is customized as required.

2.1.7 ILA:

The Integrated Logic Analyser is used for monitoring the internal signals in the FPGA. This IP consist of input probes which can be mapped or connected to the required signal for detection. It can be used in verification and debugging applications. The number of probes can be selected by choosing the customize IP option.

3. DESIGN FLOW:

Loopback Reception and Transmission of ethernet data is implemented in this design. Ethernet data received in the FPGA can be verified using the Wireshark Tool. This received ethernet data coming from the PC, is buffered into a FIFO and stored into a Block RAM. The data stored in Block RAM is given into the Ethernet controller for Transmission. This transmitted data will be received by the FPGA and again transmitted. Thus, a loopback operation is achieved. In the block design ILA cores as added, in which the waveforms and signals can be analysed. The clocking signals for each core is generated by the clocking wizard. ILA is operated with a clocking frequency of 300MHz.

4. IMPLEMENTATION:

The design is implemented using the Artix-7 AC701 FPGA board and is created using the above mentioned IP cores in Xilinx Vivado IP integrator tool. Block level design is used, which makes the IP interconnections easier and simpler. The control logic for the IP cores is written in VHDL and is added as a block to the IP integrator. The data is sent along with the UDP/IP header. These are added to the design as Blocks.

The port connections are made followed by synthesis and implementation and then bitstream is generated. This is programmed onto the FPGA board. Ethernet cable is connected between the PC and FPGA for full duplex transmission and reception. The UDP sender done in LabView is made to run, and program is loaded to FPGA and run. Figure 6 shows Block design implemented in Vivado IP integrator (including the control logic for reception and transmission).

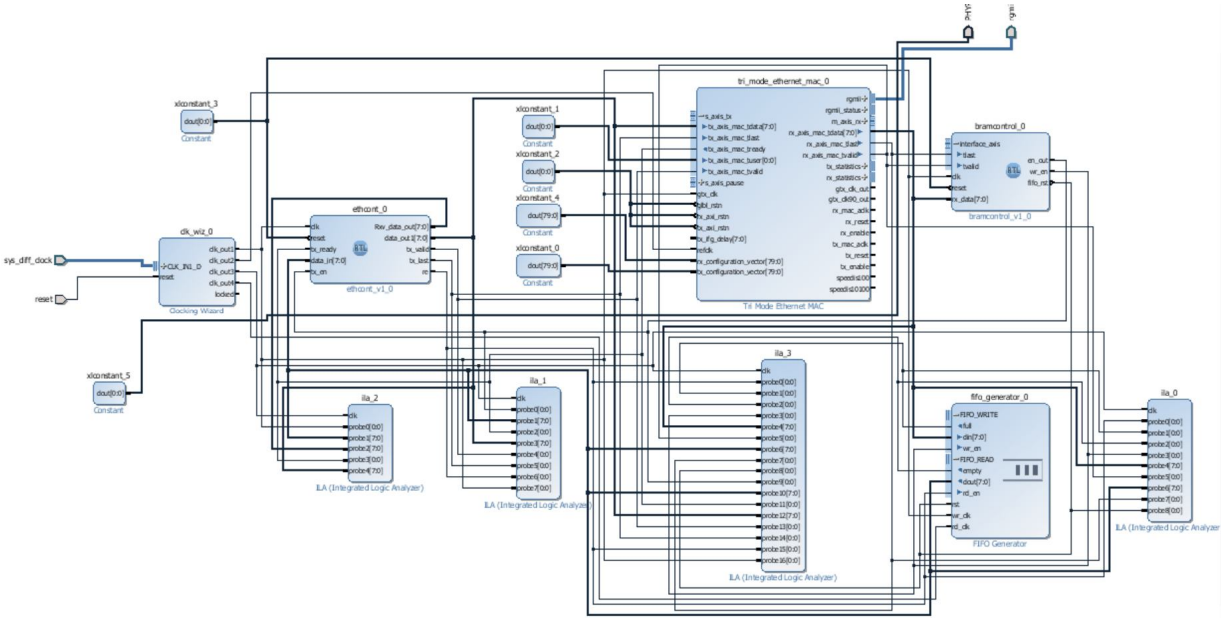

Figure 6 Block Design of the System in IP Integrator

5. RESULTS:

The UDP sender sends data from LabVIEW and is received by the TEMAC IP core in FPGA. The data is stored in FIFO and BRAM. This stored data is read from the BRAM and is transmitted by the TEMAC IP core in the FPGA. The transmitted and received packets can be viewed in WireShark. Figure 7 shows the reception of ethernet data. The received data is simultaneously written into the FIFO. When the last data is received a tlast signal indicates the end of data and enables the FIFO to read. The data is then read out of the FIFO. Figure 8 shows the transmission of the received data. The received data is taken from the FIFO and given to the transmitter control block. The data is trasmitted to another IP address. Signal waveforms are triggered in the ILA, which is shown in Figure 9. At 1261 µs, the last data is received. After 13µs (1274µs), the FIFO starts reading the data. Then the data is transmitted along with the UDP header, immediately when it receives the *en_out* signal from the controller without any delay.

6. CONCLUSION:

The implementation of reception and transmission of ethernet data is discussed. Here the IP integrator tool is used to implement the design. This tool enables the interconnection of IPs and complex signals easily than a RTL coding based design.

IPASJ International Journal of Electronics & Communication (IIJEC)
Web Site: http://www.ipasj.org/IIJEC/IIJEC.htm
A Publisher for Research Motivation ..... Email:editoriijec@ipasj.org
Volume 7, Issue 4, April 2019 ISSN 2321-5984

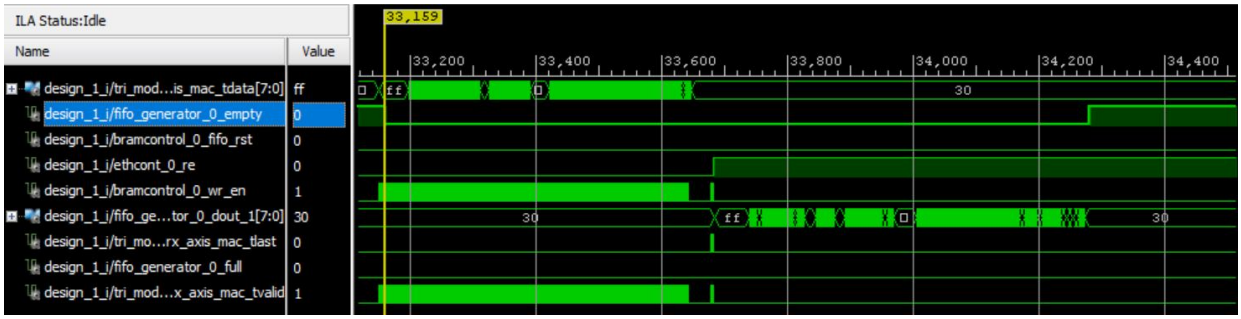

Figure 7 Ethernet Reception and Storing To FIFO Obtained in an ILA

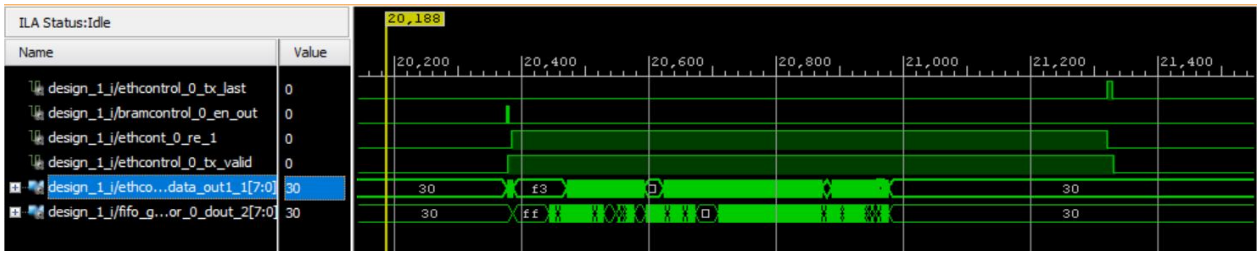

Figure 8 Ethernet Transmission Of the Received Data

Figure 9 Combined Waveform of Reception and Transmission

The IP are added as blocks and the control signals are given to make the corresponding IPs to function. Initial data to be received is sent from a UDP sender block in LabView. The received and transmitted data can be viewed in ILA or Wireshark.

7. FUTURE WORK:

Further enhancements can be made in the design of this ethernet telemetry system, like adding a ping pong buffer (with multiple FIFO and BRAM) to increase the processing speed of the system, and filtering of the received ethernet packets corresponding to the devices. Ping pong buffer is analogous to the pipelining method in processors.

REFERENCES:

[1] https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_1/ug994-vivado-ip-subsystems.pdf
[2] A. Sasi, S. Saravanan, S. R. Pandian and R. S. Sundaram, "UDP/IP stack in FPGA for hard real-time communication of Sonar sensor data," *2013 Ocean Electronics (SYMPOL)*, Kochi, 2013, pp. 1-6. doi: 10.1109/SYMPOL.2013.6701940
[3] https://www.xilinx.com/support/documentation/ip_documentation/tri_mode_ethernet_mac/v9_0/pg051-tri-mode-eth-mac.pdf
[4] https://www.xilinx.com/support/documentation/ip_documentation/fifo_generator/v13_2/pg057-fifo-generator.pdf
[5] https://www.xilinx.com/support/documentation/ip_documentation/blk_mem_gen/v8_3/pg058-blk-mem-gen.pdf
[6] A. S. Tanenbaum, Computer Networks, 4th ed. Prentice Hall, August 2009.
[7]F. L. Herrmann, G. Perin, J. P. J. de Freitas, R. Bertagnolli and J. B. dos Santos Martins, "A Gigabit UDP/IP network stack in FPGA," *2009 16th IEEE International Conference on Electronics, Circuits and Systems - (ICECS 2009)*, Yasmine Hammamet, 2009, pp. 836-839
[8] Volnei. A. Pedroni, "Circuit Design and Simulation with VHDL", The MIT Press 2010.