TRANSFER OF UT INFORMATION FROM FPGA THROUGH ETHERNET INTERFACE

C.Suneetha[1], S.Deepa Rani[2]

[1]PG Student [DECS], Dept of ECE, Gokula Krishna College of Engineering, Sullurpet, AP, India
[2]Associate Prof, Dept of ECE, Gokula Krishna College of Engineering, Sullurpet, AP, India

Abstract

*FPGA introduced in 1980's has been replaced the expensive ASIC products in the real time data processing systems. This paper presents a design for an Ethernet-based transfer of UT information to all the sub-systems on **Xilinx Spartan™-3E Starter Board** through any existing LAN can be employed to maintain timing synchronization of all the systems. A VHDL software application is developed that can communicate with the board. IRIG-B (B120) time coding format received from GPS receiver. All the systems need to communicate with external world. FPGA has work in real time. So, a high speed reliable network called Ethernet is used. In its physical layer we use media independent interface (MII). UDP a network protocol used to implement form physical to transport layer. For real time application timing analysis is done.*

Keywords: *UT, IRIG-B, UDP, MAC, MII, EHERNET, VHDL.*

--***--

1. INTRODUCTION

This project enables software and hardware modeling of Xilinx XC3S1600E kit board in system level to analyze the time taken by each sub-system to complete its operation in synchronous to the universal time (UT) information. The UT that is in IRIG-B format produces 100pps time information is transmitted as 3.3v to the kit. Now the purpose of this project is to check how many sub-systems completed their operation in time and we can also estimate intermediate operation of each system. Timing information is transmitted through Ethernet. VHDL can be used as a stimulus definition language as well as a hardware description language to design and implement. By using VHDL we can model a test bench. FPGA board could receive UT information depending on this it transmits UDP packets to clients through Ethernet. The physical layer is described by IPV4 protocol. The next is link layer, Ethernet MAC manages flow control, error checking and does CRC checking

2. DESCRIPTION OF IRIG-B

The Inter-Range Instrumentation Group (IRIG), part of the Range Commanders Council (RCC) of, originally developed the IRIG time codes and has been revised several times by the Telecommunications and Timing Group (TTG) of the RCC. The latest version is IRIG standard 200-04, "IRIG Serial Time Code Formats," updated in September 2004. In order to ensure that all the sub-systems in an entire system orderly working synchronously on time scales. Hence high precision time system equipment is a key part of the system. The extensive use of GPS receiver can produce UT information.

2.1 Available Formats

Although the "IRIG-B" time code is best known, the standard actually defines a family of rate-scaled serial time codes. The six code formats use different pulse rates, or bit rates, as shown in the Table given below:

Table-1

Format	Pulse rate (or Bit rate)	Index count Interval
IRIG-A	1000PPS	1ms
IRIG-B	100PPS	10ms
IRIG-D	1PPM	1min
IRIG-E	10PPS	100ms
IRIG-G	10000PPS	0.1ms
IRIG-H	1PPS	1second

From the above formats to implement this project I selected IRIG-B which has a bit rate 100pps and index count interval of 10ms. [1]

The IRIG-B format used in this project is **B120**
B ----------- Format IRIG-B
1 ------------ Sine wave amplitude modulated
2 ------------ 1Khz Carrier Frequency / 1ms Resolution
0 ----------- BDCTOY, CF, SBS.

This paper implements NTS that uses GPS as time synchronizing source to have precise, accurate, reliable and stable time. A GPS receiver is used to receive time information. By using the above IRIG format time code logic has been generated and decoded. The logic has been designed in VHDL and implemented on FPGA. This code produces Universal Time (UT) information to the kit and this information is used to analyze the operation of clients which are connected to other end of the kit. The process is implemented by taking inputs and outputs as 100MHz clock as input signal (clk_out), active high reset (reset), UT modulated code (ut_in), time out @ 1sec (ut_data_out) and to indicate status of UT (ut_rdy_out) .

The IRIG-B time code logic is processed in two phases.

- UT Generation logic----Process to generate 1khz clock for sampling UT signal and synchronized with UT.
- UT Decoding logic---- Process to decode the UT signal and taken time information.

The below Figure is time code logic flow diagram showing of generation and decoding of clock

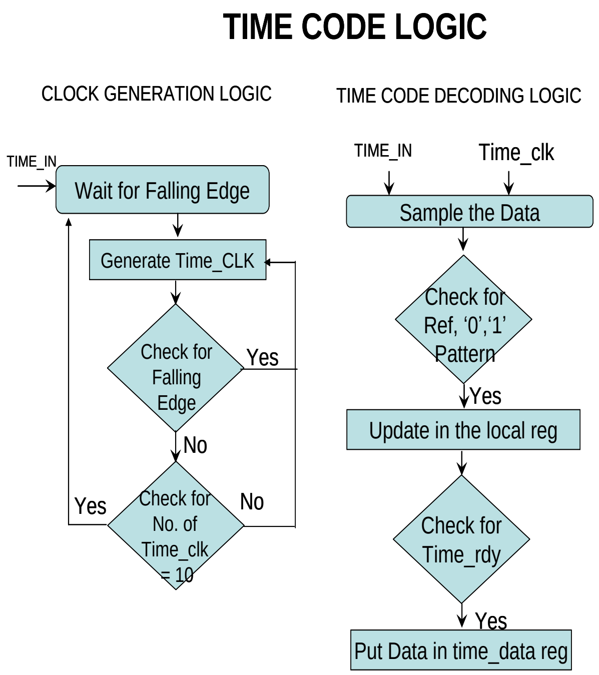


Fig 1: Time Flow Logic Diagram

3. DESCRIPTION OF UDP AND ETHERNET

UDP or the User Datagram Protocol is a best-effort transport standard that is layered on the IP protocol. UDP is suitable for purposes where error checking and correction is either not necessary or performed in application avoiding the over head of such processes this is it is used in real-time transmissions such as media streams where on-time delivery carries a higher priority than bit-correction. UDP processes employ ports to indirectly identify each other and often make use of known, reserved addresses on a per application basis[3]. UDP provides a connectionless communication between the hosts. For packetization of the signals UDP was chosen considering the fact that data is to be transmitted and received in real-time. The input signals are sampled and stored in the buffer while the UDP server opens the sockets necessary for transmission and reception.

Ethernet defines physical interconnection and link layer protocol is connected using "MII" bus. In this project Ethernet protocol is standardized by IEEE 802.3u standards. It uses 48bit MAC address space to uniquely identify device. The IP packets have an MTU of about 65,000bytes while the Ethernet has an MTU of 1500bytes. So, an IP packet cannot fit into a single Ethernet frame, therefore the IP packets are split into and sent over multiple Ethernet frames. At the end of each Ethernet frame there is a CRC32 field that is calculated from entire frame, excluding preamble and start of frame (SOF) bytes. The project which is implemented by Micro Blaze development kit board includes SMSC LAN 83C185 10/100 Ethernet PHY and an RJ-45 connector to support UDP transmission and reception to interface with spantan3E.

4. DESIGN AND IMPLEMENTATION

The task of the paper is to design FPGA to operate in real time mode. NTP is a Networking protocol for clock synchronization between computer systems over packet switched variable latency data network. NTP is one of the oldest internet protocols in use.

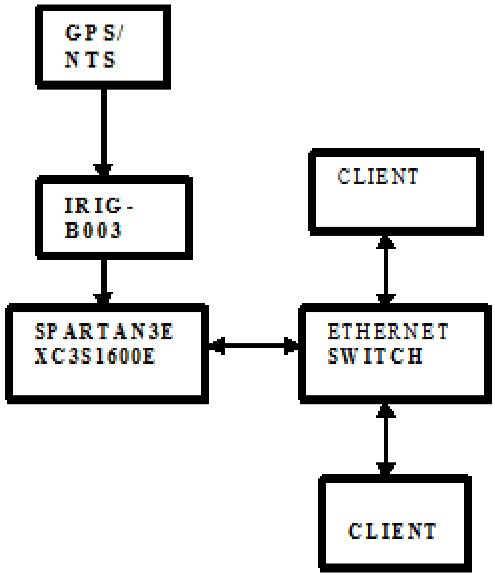


Fig 2: Block Diagram

This paper is designed as an interface that enables FPGA board to communicate with other devices via onboard Ethernet connection. The programmable logic board Xilinx Spartan 3E – 1600 development systems programmed with VHDL code. The center piece of the board is a Spartan 3E. This FPGA kit is used to transfer through Ethernet. This Micro Blaze development kit board includes SMSC LAN 83C 185 10/100 Ethernet PHY interface and used on RJ 45 connector. An ISE project contains all the files needed to design a piece of hardware and downloaded to the FPGA. When power is ON FPGA it is in configuration mode and streams of 0's and 1's are downloaded through some special pins to configure it. Now it operates in user mode. Design of an FPGA is usually "synchronous". So the IRIG –B time format coding is allowed to clock all the D flip flops to simultaneously take a new state at the same time a special global routing or global lines are provided to distribute clock all over the kit board.

5. PROCESSES AND ALGORITHMS

This project was subdivided into three main components: the Universal time module, the Receiver Module and the Transmission Module. All the components were developed simultaneously. Details are discussed in the succeeding sections.

5.1 Universal Time

IRIG time code is framed with repeating bits each containing 100 bits are numbered from 0 -99. During the start of each bit time a data packet is transmitted to the clients through Ethernet switch. At every 10th bit starting i.e. 9, 19, 29……. 99 which are identified as positions P1,, P2…P0. During this bit positions data packets are transmitted to the clients to synchronize them to universal time.

5.2 Transmitter Module

In this module the data is encapsulate with UDP, Ethernet and IP protocols. UDP/IP Ethernet IP core easily enables FPGA based subsystems to communicate with other subsystems via Ethernet, using UDP protocol. It is highly configured in Xilinx Spartan 3E FPGA architecture.

Table 2: UDP format

Source port	Destination port
Length	Check sum
Data Packets	

UDP Fields
Source port:- indicates the port to which time information data to be send.
Destination port: - Internet destination address
Length: - includes header and the data.
Checksum: - 16 bit one's compliment
The protocol IPV4 is that for UDP is 11.

Ethernet Addressing

It supports the common media independent interface (MII) with 4 bit wide transmit and receive interfaces running at 125 mega hedges and wired at a speed of 100 Mbps.

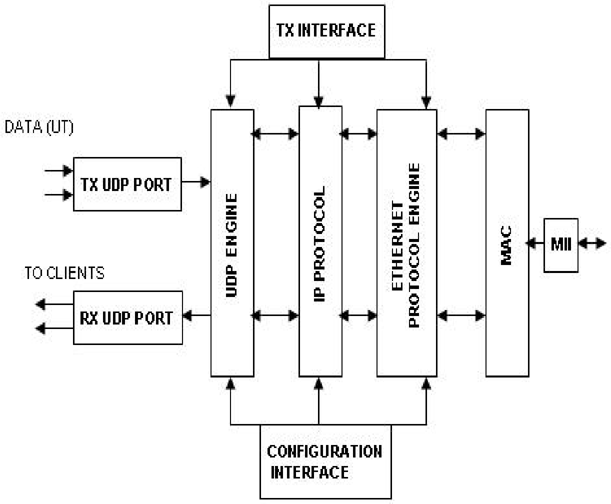


Fig 3: Ethernet addressing

By using a PC that serves as a central server is interfaced with this board and the other end of FPGA all the clients are connected. The data communication is via Ethernet of IEEE 802.3u standards that enable many hosts to transmit and receive data over a twisted pair network. I initially worked on sender module to determine local address host name packet size. The UT time is activated and then UDP server is activated. The sample of the input signal is encapsulated by the UDP. The packetized data is stored in the buffer. UDP is chosen considering the fact that data is to be transmitted and received in real time.

5.3 Receiver Module

The module is developed in different stages. The process is analyzed by the IP port address, start of frame (SOF), UDP source port and destination port and Ethernet payload. Initially in this module the clients sends a request to FPGA to transmit Universal Time (UT) data. The data packets are then sent out through the Network Interface Card of the PC to the Local Area Network (LAN). Over the Ethernet, the packets are sent through the router to the FPGA receiver connected to the LAN through its onboard RJ-45 connector. The FPGA board then performs de-packetization and conversion of timing packets back to analog signals using its On board digital to Analog convertor (DAC)

6. TESTS AND TEST RESULTS

To implement this project all the modules are coded in VHDL language and tested its functionality by simulating first. After simulation each and every module is tested on FPGA. The Timing Information of UT and sub system timings are analysed on chip Scope pro analyser.

Basic Test:

In the first Experiment, we connected a simple GPS receiver. A VHDL program is written to produce a clock for UT information.

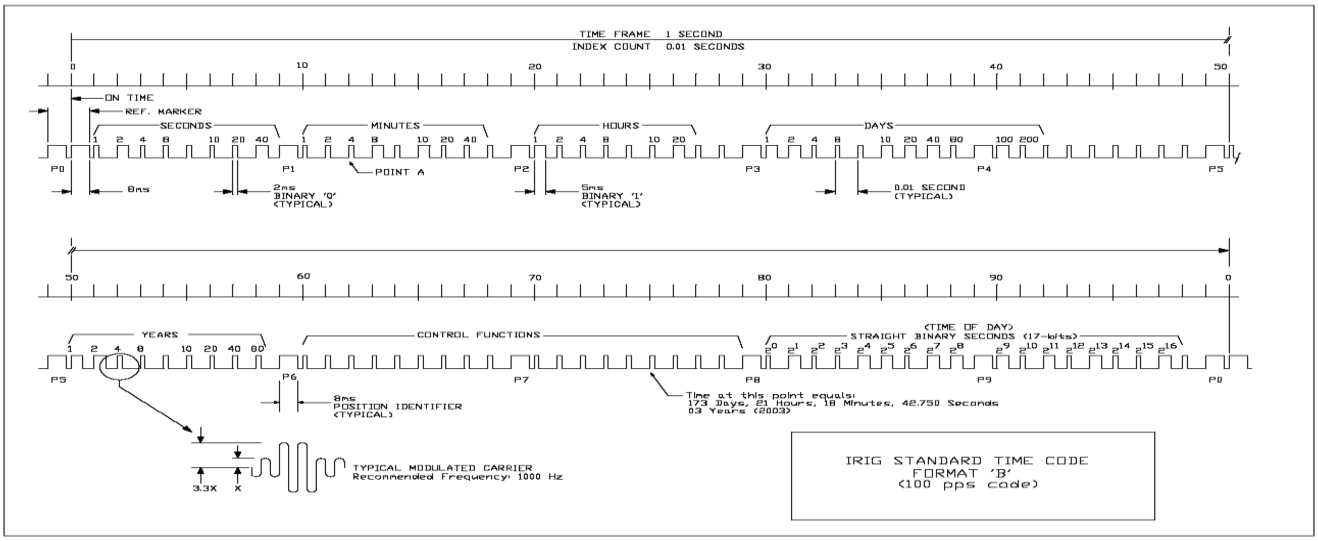
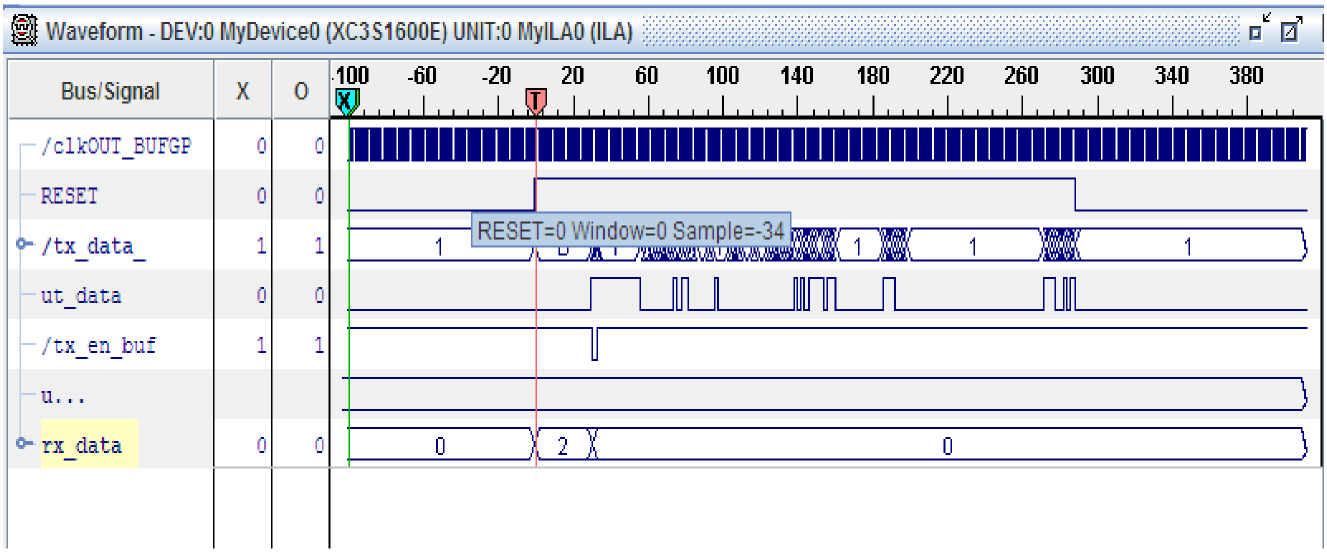


Fig 4: Output of UT

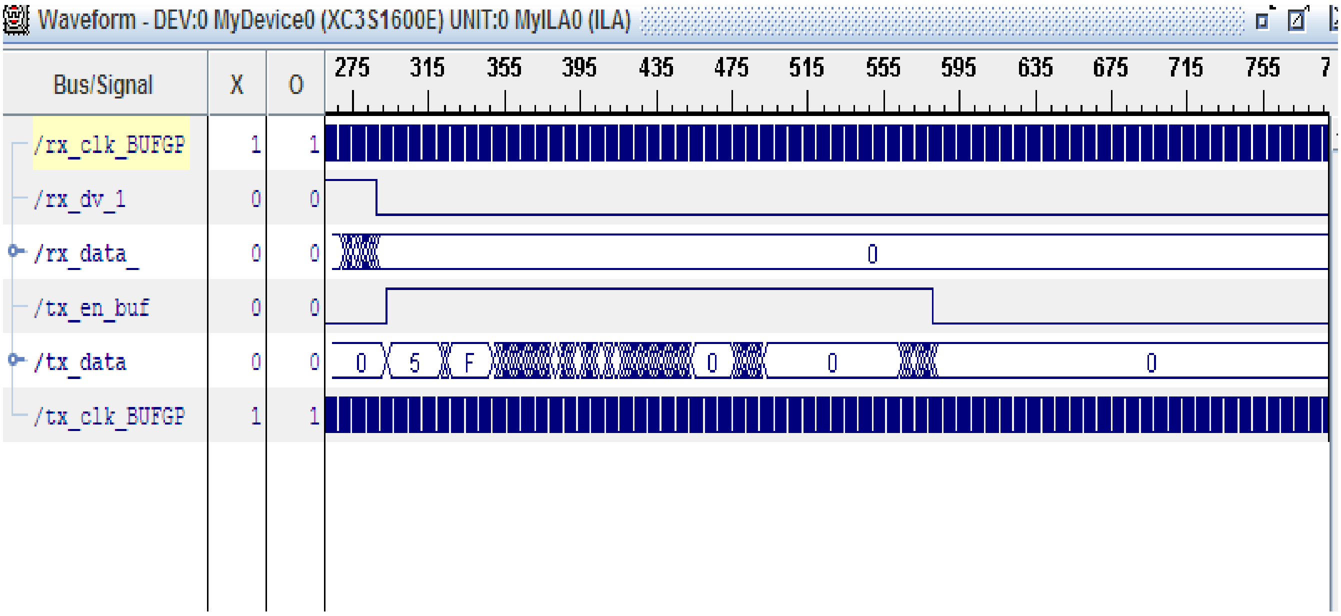
Chip Scope Outputs of Transmitter and Receiver Modules are given below:



Waveform - DEV:0 MyDevice0 (XC3S1600E) UNIT:0 MyILA0 (ILA)

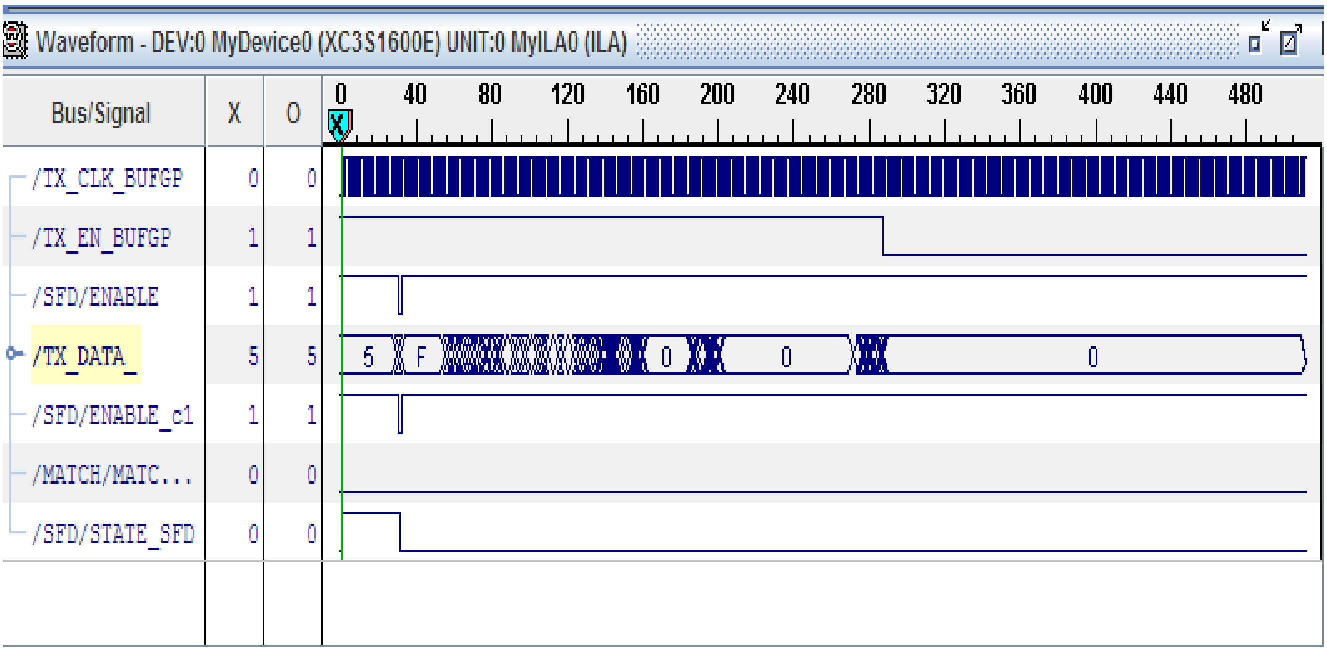
Bus/Signal	X	O
/clkOUT_BUFGP	0	0
RESET	0	0
/tx_data_	1	1
ut_data	0	0
/tx_en_buf	1	1
u...		
rx_data	0	0

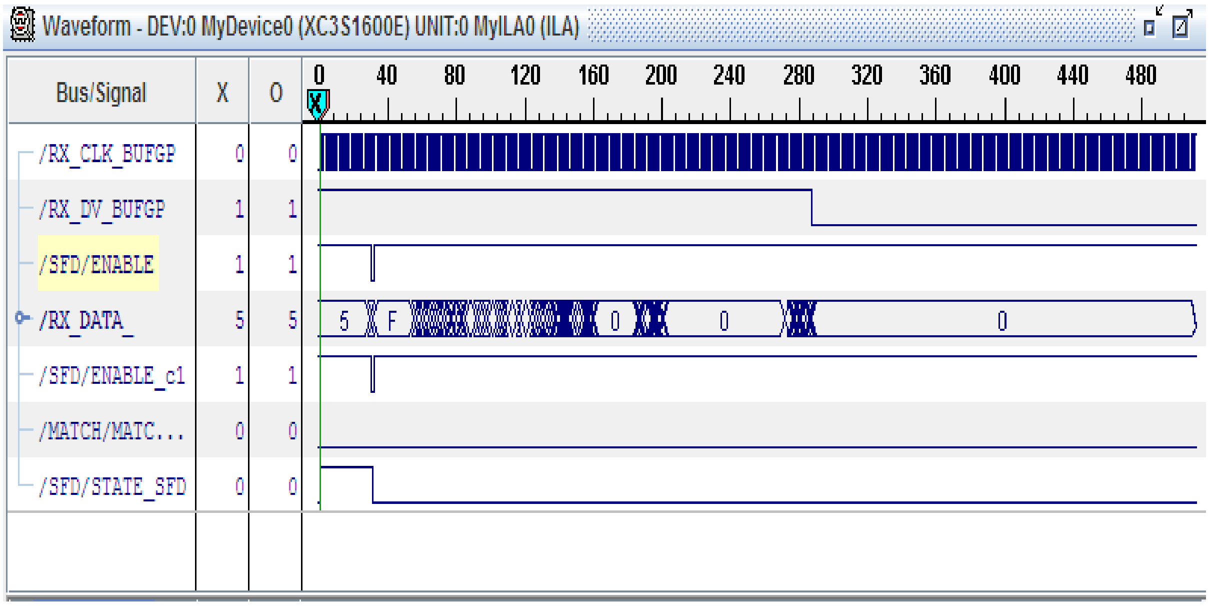
RESET=0 Window=0 Sample=-34



Waveform - DEV:0 MyDevice0 (XC3S1600E) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O
/rx_clk_BUFGP	1	1
/rx_dv_1	0	0
/rx_data_	0	0
/tx_en_buf	0	0
/tx_data	0	0
/tx_clk_BUFGP	1	1





7. CONCLUSIONS

The purpose of this paper was to design and built an Ethernet based programming of Xlinx Spartan 3E development board to transmit Universal time information as data packets using UDP/IP and receiving clients request for synchronization with universal clock. Based on the tests done on the system the three modules are simulated synthesized and analyzed through chips scope pro analyzer. The future scope of this paper provides a more generic implementation of all versions of FPGA devices by any vendor. An appropriate hand shaking protocols can be developed for automatic packet loss checking and retransmission. Bulk data can also be transferred using memory storage devices under proposed core. By using Xlinx Spartan 3E series, the cost of implementation reduces to a great extent.

REFERENCES

[1]. www.cyber-sciences.com
[2]. handbook@javvin.com
[3]. Steinke.netwrok tutorials 5th Edition CMP books.
[4]. http://opencores.org.
[5]. http://www.FPGA4fun .com/100BASE-T3.html
[6]. UG 257 Micro Blaze development Spartan 3E 1600E
[7]. SMSC LAN83C185 Data Sheet
[8]. www.xilinx.com/edk
[9]. http://www.xilinx.com/support/download/index.htm