Implementation of PTP Stack on FPGAs

Sindhu R, Suchita Kumari, Rajani K, Rakshita Y N, Saravana Kumar

***Abstract*— Precision Time Protocol, described in IEEE 1588 is able to synchronize distributed clocks with an accuracy of less than 1 microsecond. Precision Time Protocol's operating principle is to exchange messages consecutively to determine the offset between master and slave. The precision and the performance of Precision Time Protocol is based on the precision of Timestamp. The timestamp of communicating packets achieve synchronization between two devices. In this paper, we are implementing the Precision Time Protocol stack on FPGAs to show the working of Protocol and the different layers of OSI model. Precision Time Protocol stack is developed by implementing MAC layer with VHDL code and the upper layer with C code. The developed stack on FPGA acts as Master used to communicate with Slave (in our case, Computer) and a communication is established.**

***Index Terms*— Clock; Synchronization; PTP; One step clock; Grandmaster clock; Slave clock**

I. INTRODUCTION

We use clocks to synchronize ourselves with persons or any processes. Synchronization becomes a necessity when devices are working at a distance from each other. Not every clock is sufficiently accurate. Two effects are in evidence when setting or synchronizing clocks: Two effects are need to be observed when setting or synchronizing clocks: independent clock initially run at an offset for one thing. To synchronize them, the inaccurate clock is set to the more accurate one. Another thing is that real clocks do not run at the same speed. Therefore, the speed of more accurate clock has to be regulated constantly. In such cases, a Master clock synchronizes with the slave clock networked within the same system. Due to this need for synchronization, PTP was released as a standard of protocol in 2002. However, if two clocks are at the same rate, there is no guarantee that they will remain in synchronization. This is why the process of synchronization is continous. Several factors can cause two identical clocks to lose synchronization such as differences in temperature, age of clock, and the rate of frequency. Synchronization is needed because of these factors.

Manuscript received May 16, 2015

Sindhu R, Dept. of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, India

Suchita Kumari, Dept. of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, India

Rajani K, Dept. of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, India

Rakshita Y N, Dept. of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, India

Saravana Kumar, Associate Professor, Dept. of Telecommunication Engineering, Dayananda Sagar College of Engineering, Bangalore, India

II. IEEE 1588 -2008 SYNCHRONIZATION PRINCIPLE

Fig. 1 shows one step clock.

The one-step clock technique exchanges three packet-sized messages between master and slave. The grandmaster clock periodically sends a packet called a 'sync' packet containing a timestamp. A slave clock receives the grandmaster's 'sync' packet and its arrival time. The slave clock sends Delay_Req message to the grandmaster clock. Then, the grandmaster clock sends Delay_Resp massage to the slave clock. By sending and receiving these synchronization packets, the slave clock can accurately measure the offset between their local clock and the master's clock. The slave can then adjust their clocks by using offset to match the time of the master clock. The master stamps the time (t1) on the sync massage that leaves the master. The slave receives the sync message at time (t2) and sends a Delay_Req message at time (t3) to the master, which the master receives at t4. The master then sends Delay_Resp packet to the slave.

Slave time offset= ((t2–t1)–(t4–t3))/2
(1)

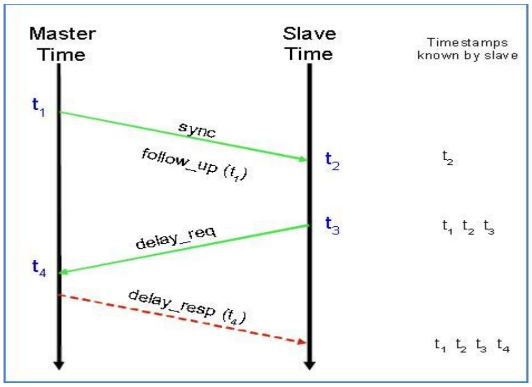

Fig 1 One step clock

III. PTP STACK BASICS

PTP stack needs to be developed in order to implement PTP. The PTP stack may be used to act as an IEEE 1588 master/slave within a PTP network.PTP is an application layer protocol, on top of UDP/IP/Ethernet. The UDP port number identifies UDP datagram as PTP message. In PTP stack, the UDP encapsulates the upper layer PTP messages in the stack. To detect PTP packet in UDP layer, port number 319 and 320 are used. The use of port number 319 is reserved for messages that have to be timestamped such as sync message and delay_req message. All other messages use port number 320

for delay_resp message. PTP message encapsulated into UDP datagram is encapsulated in IP datagram and again this is encapsulated into a MAC frame and then transmitted.

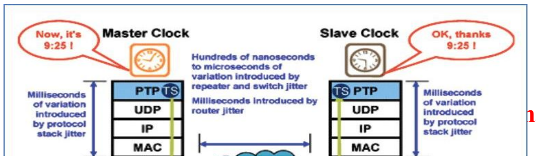
Now, it's 9:25 !
Master Clock
Slave Clock
OK, thanks 9:25 !
Hundreds of nanoseconds to microseconds of variation introduced by repeater and switch jitter
Milliseconds introduced by router jitter
Milliseconds of variation introduced by protocol stack jitter
PTP TS
UDP
IP
MAC
TS PTP
UDP
IP
MAC
Milliseconds of variation introduced by protocol stack jitter

Fig.2 PTP stack

PTP's operating principle is to exchange messages to determine the offset between master and slave and also it determines the message transit delay within a network.

IV. ESTABLISHING COMMUNICATION BETWEEN PTP MASTER AND SLAVE

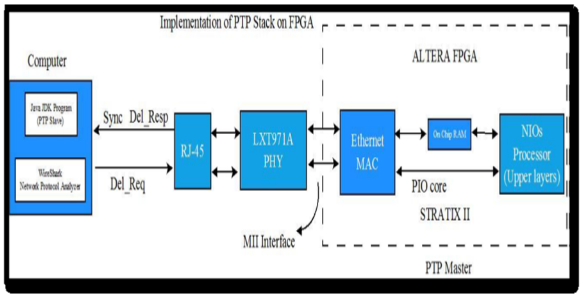
Implementation of PTP Stack on FPGA
Computer
ALTERA FPGA
Java JDK Program (PTP Slave)
WireShark Network Protocol Analyzer
Sync Del_Resp
Del_Req
RJ-45
LXT971A PHY
Ethernet MAC
On Chip RAM
NIOs Processor (Upper layers)
PIO core
STRATIX II
MII Interface
PTP Master

Fig.3 Communication between PTP Master and Slave

Communication is established between PTP master and the slave and analyse the protocol analyse the protocol as well as data packet transfer using Wireshark software. Wireshark is free and open source packet analyzer, used for network troubleshooting, software and communication development protocol development. Wireshark is a network packet analyzer used to capture network packet and tries to display data as detailed as possible. In Wireshark, we can observe the packets and type of packets being transferred in and out from the computer. This Network Protocol Analyser ensures the communication been established and also type of message been communicating.

V. ANALYSIS OF NETWORK PACKET

After the communication is established, the network packet can be analyzed using Wireshark software. The screenshots of the Wireshark shows the network packet received and sent from the computer. In the screenshots that follows, we can observe different packets like sync packet, delay_req packet, delay_resp packet, ICMP (ping), source IP address and destination IP address.

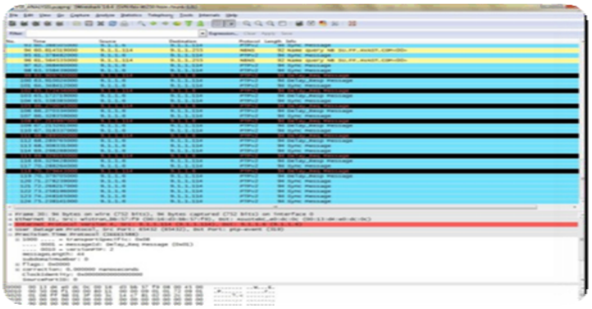

Fig. 4 Sync, Delay_Req and Delay_Resp messages

In Fig.4, we can observe that sync message is sent from master every second. Master replies to slave by sending Delay_Resp when Delay_Req is sent from the slave.

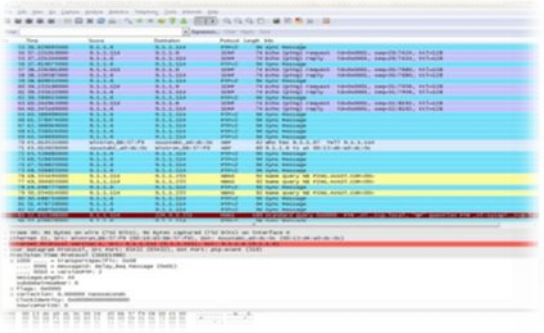

Fig.5 Echo Request and Echo Reply Messages

CONCLUSION

The PTP is used to synchronize distributed clocks with an accuracy of less than one microsecond. To achieve accuracy of less than one microsecond, it is necessary to design an implementation of the PTP Protocol. In this paper, OSI layer is implemented on Altera FPGA, in which PTP is an application layer protocol. The PTP stack acts as an IEEE 1588 master/slave within PTP network. The PTP stack on FPGA is the master used to communicate with the slave and a communication is established used for clock synchronization of slave.

FUTURE WORKS

IEEE 1588 has a wide acceptance on the automation and measuring technology market but the methods are also of interest to many other fields. To meet additional requirements, the IEEE1588 v2 project was started in February 2005 in the committee with the aim of extending the standard in the following directions: Greater accuracy, Fail safety, Simple Network Management Protocol (SNMP) management, various Ethernet headers, transparent clock, Application in other network types and security.

REFERENCES

[1] IEEE 1588/2008, Standard for a precision clock synchronization protocol for Networked Measurement and Control Systems.

[2] Precision clock synchronization-IEEE 1588 White Paper Rev.1.2.

[3] Weibel, Hans, Beehaz, Domnic, "IEEE 1588 Implementation and Performance of Timestamping Techniques", 2004 conference on IEEE 1588, 2004.

[4] [4] www.wireshark.org

[5] John C.Eidson, IEEE-1588 Version 2, Agilent Laboratories, Measurement Research Lab, March 5, 2007.

[6] Thomas Müller, Alexander Ockert, Hans Weibel: "PHYs and Symmetrical Propagation Delay", 2004 Conference on IEEE 1588, Sept. 27-29, Gaithersburg