Design and Implementation of IEEE1588 Time Synchronization Messages Timestamping Based on FPGA

Mingzhu Qi, Xiaoli Wang*
School of Mechanical, Electrical & Information Engineering
Shandong University at Weihai, Weihai
Weihai, P. R. China
q-mingzhu@hotmail.com, wxl@sdu.edu.cn

Zhiqiang Yang
Integrated Electronic Systems Lab Co. Ltd.
Jinan, P. R. China
yangzhiqiang@ieslab.com.cn

Abstract—**IEEE1588 is a Precise Time Protocol (PTP), which is of potentially wide application in control and measurement networks. Stamping PTP messages accurately in physical layer taking advantage of hardware circuits, it is one of important key technologies to achieve the object of high precision time synchronization of IEEE1588. This paper analyzes the content of IEEE1588 standard in detail, and proposes a method of "Cyclone II FPGA + MAC + PHY", a hardware solution that stamps the time of PTP messages' arrival and departure. It permits to obtain messages from Media Independent Interface (MII) between MAC and PHY layers by using "DM9000 + LXT971A", while FPGA monitors these signals and timestamps synchronization messages. To validate the solution, a test platform was built. The result shows that the solution can detect the sent and received synchronization messages accurately. This work laid a solid foundation for developing high-precision network clock based on the IEEE1588 standard.**

Keywords-IEEE1588; clock synchronization; Media Independent Interface (MII); FPGA

I. Introduction

Along with the development of the electrical power supply system, more and more intelligent electronic devices for measurement, protection and control have been put into use to improve the stability and reliability of the system. But there is also an increasing demand on accurate time synchronization which trails close behind.

The infamous blackout from August 2003 in the Northeast United States and Ontario, Canada, had the side effect of serving as another stimulus for the time synchronization issue. Much effort was needed in the post-fault analysis to align the fault records from different measurement locations, because the time synchronization of the recording devices was poor [1] [2]. This sounded the alarm, and makes time synchronization become a focus. Nowadays, Ethernet network has the potential of becoming the communication backbone of those equipments, but it is not kind of real-time network. Time synchronization in those networks appears to be especially meaningful. As is well-known, Simple Network Time Protocol (SNTP) was already specified in IEC61850 as time synchronization mechanism [3] [4], but it applies only to low precision. For the more complex applications, IEEE1588 seems to be the right choice. Thus, this paper researches into the protocol of IEEE1588, and according to the characteristics of IEEE1588, a hardware solution that stamps the time of PTP messages' arrival and departure was designed. In addition, the solution was simulated in Modelsim.

II. The Basic of IEEE1588

IEEE 1588 is a master-slave synchronization protocol. There are two main functions established by the IEEE 1588 standard. Master clock and slave clock from the network continue to exchange information, which determines the time offset and link delay between them. And the slave clock uses time information to adjust its own local clock in order to achieve the synchronization of the reference precise clock [5]. The level of accuracy based on timestamp, timestamp can use software or hardware source. The advantage of this approach is use of hardware. It permits high accuracy time synchronization, real-time is guaranteed. What's more, it can directly use TCP/IP protocol in communication.

A. The Type of PTP Messages[6]

IEEE1588 defines event and general PTP messages. Event messages are timed messages in that an accurate timestamp is generated at both transmission and receipt, such as Sync, Delay_Req, Pdelay_Req, and Pdelay_Resp. General messages do not require accurate timestamps, such as Announce, Follow_Up, Delay_Resp, Pdelay_Resp_Follow_Up, Management, and Signaling.

The Sync, Delay_Req, Follow_Up, and Delay_Resp messages mentioned above are used to generate and communicate the timing information needed to basic synchronize clocks.

B. The Way of Time Synchronization

The basic time synchronization process is divided into two phases as Fig. 1 shows:

1st step: master clock node sent the messages of Sync and Follow_Up periodically (usually 2s), and recorded the transmission time of Sync message (T_1) which would be encapsu-

*Corresponding author: wxl@sdu.edu.cn.

This work was supported by the Independent Innovation Foundation of Shandong University (No. 2010ZRJD001).

lated into Follow_Up. And slave clock node recorded the received time of Sync (T_2).

2nd step: the slave sent Delay_Req messages periodically (usually 4-60s), and recorded the transmission time of Delay_Req (T_4), Then the master received Delay_Req, and recorded the time of receipt (T_3), which encapsulated into Delay_Resp later. Then Delay_Resp message returned to the slave.

As a result, the slave obtained T_1-T_4, and these points could help to calculate time offset and link delay as follow:

$$Offset + T_2 = T_1 + Delay \tag{1}$$

$$T_3 = T_4 + Offset + Delay \tag{2}$$

Then,

$$Offset = T_1 - T_2 + Delay \tag{3}$$

$$Delay = (T_2 - T_1 + T_3 - T_4)/2 \tag{4}$$

The slave uses Offset and Delay to achieve time synchronization.

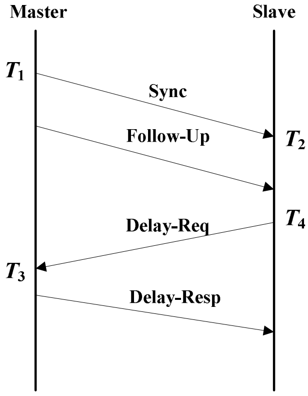

Figure 1. The process of time synchronization.

III. Hardware Solution

A. The Structure of Hardware Platform

IEEE1588 specifies a particular point for stamping the time of PTP messages' arrival and departure. This point may be in the application layer, in the kernel or interrupt service routines, or in the physical layer. In general, the closer this point is to the actual network connection, the smaller the timing errors introduced by fluctuations in the time taken to traverse the lower layers. As for this solution, it is designed that timestamps are generated between the media access control (MAC) layer and the physical layer (PHY) where is usually called MII.

Fig. 2 below shows the basic structure of the platform. CPU as the main processor, which S3C2440A produced by SAMSUNG is a good choice, is used to establish the mechanism of PTP message synchronization, and manage the operation of this system. MAC and PHY shall be both single chip, and Connect through the MII interface. Details are described below. FPGA is the key component for timestamp, which shall be programmed to detect messages from the MII interface and feed timestamps back to CPU through BUS, here the series of Cyclone II FPGA is selected.

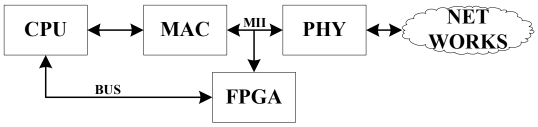

Figure 2. Hardware frame-chart.

B. The Design of Network Module

1) Chip Selection

a) DM9000[7]: DM9000 is a fully integrated and cost-effective single chip Fast Ethernet MAC controller with a general processor interface, a 10/100M PHY and 4K Dword SRAM. It is designed with low power and high performance process that support 3.3V with 5V tolerance. What's more, it provides a MII interface to connect other devices that support MII interface.

b) LXT971A[8]: LXT971A is a single-port Fast Ethernet 10/100 transceiver that supports 10 Mbps and 100 Mbps networks and complies with all applicable requirements of IEEE 802.3. It provides both an MDIO interface and a Hardware Control interface for device configuration and management. In the MDIO Control mode, the LXT971A can read the Hardware Control interface pins to set the initial values of the MDIO registers. Once the initial values are set, bit control reverts to the MDIO interface.

2) The Desgin of Non-integrated Network

DM9000 and LXT971A are chosen to construct the network module. MII interface from them can directly connect in Fig. 3. As for LXT971A, MDDIS should be set low, then MDIO port is enabled for both read and write operations and the hardware control interface is not used. Then, DM9000 can control LXT971A through MDIO interface.

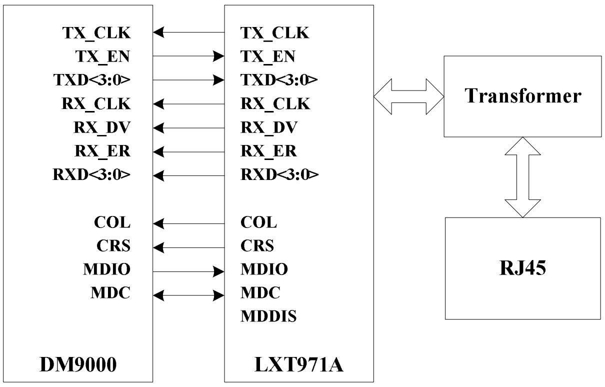

Figure 3. Network module.

IV. The Software Design of FPGA Module

The software design of FPGA plays an important role for the hardware solution. Its job is to monitor messages, stamp the time points, and feedback to the processor. This paper will focus on the analysis of messages.

A. The Analysis of MII Signals

The purpose of MII interface is to provide a simple, inexpensive, and easy-to-implement interconnection between MAC sublayers and PHYs for data transfer [9]. And the MII interface incorporates four sets of signals: Transmit channel signals, Receive channel signals, Network status and Management.

- Transmit channel signals are used to transmit data from the MAC: TXD<3:0> (transmit data), TX_CLK (transmit clock), and TX_EN (transmit enable). TXD is a bundle of 4 data signals that carry messages. TXD and TX_EN shall transition synchronously with respect to the TX_CLK.
- Receive channel signals are used to pass received data to MAC: RXD<3:0> (receive data), RX_CLK (receive clock), RX_DV (receive data valid), RX_ER (receive error). RXD is also a bundle of 4 data signals as TXD similar. RXD, RX_DV, and RX_ER are synchronous to RX_CLK.
- Network status signals are used to indicate the status of network: CRS (carrier sense) and COL (collision detected). CRS shall be de-asserted by the PHY when both transmit and receive media are idle. COL shall remain asserted while the collision condition persists. Both of them are not required to transition synchronously with respect to either the TX_CLK or the RX_CLK.
- Management signals are used to transfer control information and status: MDC (management data clock) and MDIO (management data input/output). MDIO is driven by the PHY synchronously with respect to MDC. These signals refer to the management of MII interface, and unnecessary for detecting messages.

To conclude, the signals of Transmit channel, Receive channel and Network status need to be listened to obtain PTP messages, In addition, the MII, transmission and reception of each octet of data shall be done a nibble at a time with the order of nibble transmission and reception.

B. The Analysis of PTP Message Formats

According to the provisions of IEEE1588, PTP messages may be mapped to User Datagram Protocol (UDP). And for MII, MAC frames are actually listened.

1) MAC Frame Formats: Fig. 4 shows the format of MAC frame [8]. And there are four basic elements that describe as follow, whether PTP messages will be obvious.

a) Preamble: The preamble shall consist of 7 same octets as 10101010.

b) Start Frame Delimiter(SFD): SFD indicates the start of a frame and follows the preamble. Its value is 10101011. The last nibble of this field is used as timestamp.

c) Protocol: Protocol is one of the IP header fields. It indicates the upper layer protocol carried by payload. When the upper layer uses UDP protocol, its value is 0x11 [10].

d) UDP Destination Port(UDP Dst Port): The UDP destination port of an event message(Sync or Delay_Req) shall be 319[6].

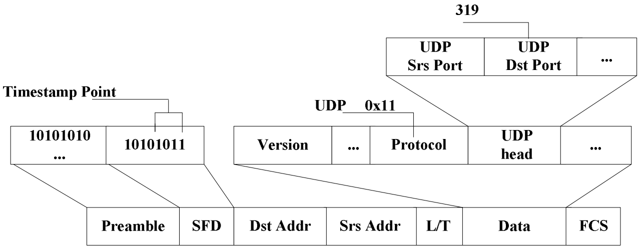

Figure 4. MAC frame format.

2) PTP Message Formats: All messages shall have a header, body, and suffix. PTP messages have the same general header. Tab. I shows common message header of PTP messages [6]. There are also two key points to identify messages.

a) messageType: The value of messageType shall indicate the type of the message as defined. If its vaule is 0 or 1, the message is Sync or Delay_Req.

b) sequenceId: The value of the sequenceId field shall be assigned by the originator of the message. In order to match messages, this field need to be recorded.

TABLE I. COMMON MESSAGE HEADER

Bits								Octets	Offset
7	*6*	*5*	*4*	*3*	*2*	*1*	*0*		
transportSpecific				messageType				1	0
reserved				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
Reserved								1	5
flagField								2	6
correctionField								8	8
Reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField								1	32
logMessageInterval								1	33

C. The Design of State Machine

According to the six key point mentioned before, the state machine using one-hot code is designed in Fig. 5. What's more, the transformation of states also refers to the number of nibbles through MII.

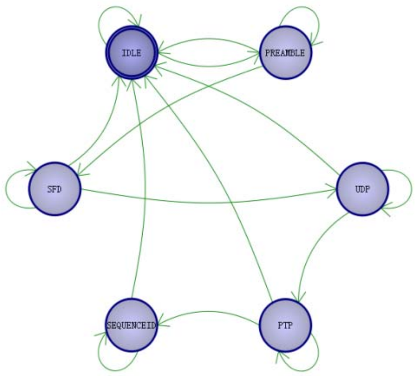

Figure 5. State machine for the module. (by Synplify Pro)

- IDLE: IDLE is the initial state. When 1010 appears, state goes to PREMBLE.
- PREAMBLE: If fifteen consecutive nibbles are all 1010, and the sixteen nibble is 1011, it indicates that SFD is detected. The trigger signal of timestamp is set, and state goes to SFD. Otherwise, return to IDLE.
- SFD: The 63 and 64 nibbles belong to the field of Protocol, and if both of them are 0001, the value of Protocol is 0x11, then the upper layer uses UDP protocol, state goes into UDP. Otherwise, return to IDLE.
- UDP: The nibbles that range from 89 to 92 belong to the field of UDP Dst Port. If the value of it is 319, the message is issued by PTP application. Then, state goes into PTP. Otherwise, return to IDLE.
- PTP: If the 102 nibble, the field of messageType, is 0 or 1, the message is Sync or Delay_Req, and the corresponding trigger is set, state goes into SEQUENCEID. Otherwise, return to IDLE.
- SEQUENCEID: In this state, Nibbles that range from 160 to 163 is recorded, and the module outputs data ready signal. When the number of nibbles beyond 165, state return to IDLE, waiting for next.

D. *The Result of Synthesis*

Fig. 6 shows the RTL schematic diagram of FPGA module. There are two module according different channel signals. The MII signals input except management signals. As for output, every module has its own output of Trigger, DataReady, message flags and sequenceId.

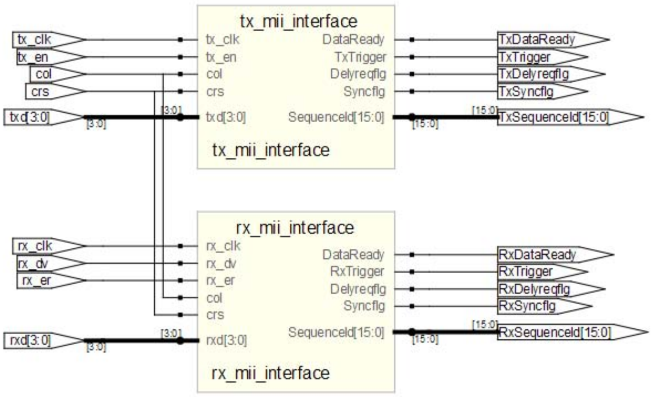

Figure 6. RTL schematic diagram. (by Synplify Pro)

V. Simulation Results

A. *Simulation Environment and Data*

Modelsim is a simulator of choice for both ASIC and FPGA design. It is one of the most popular simulation tools. In this case, Modelsim is used to simulate the module that detects PTP messages.

As for simulation, input data is important. A few of frames, with similar characteristics in Tab. II, are chosen to input the module by nibbles.

TABLE II. THE EXAMPLE OF MESSAGES

	Sync	**Delay_Req**
Preamble(7 Octets)	10101010	10101010
SFD	10101011	10101011
Protocol	00010001	00010001
UDP Dst Port	319	319
messageType	0	1
sequenceId	1	2

B. *The Result of Simulation*

Rx_mii_interface from Fig. 6 is used to explain the result through some practically typical signals studies.

Fig. 7-Fig. 11 shows the key point of a Sync frame. These results could contrast with Tab. II.

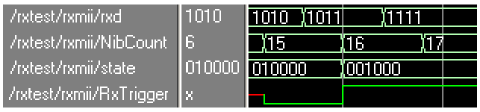

Figure 7. The signal of RxTrigger

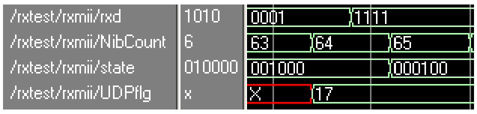

Figure 8. The signal of UDPflg.

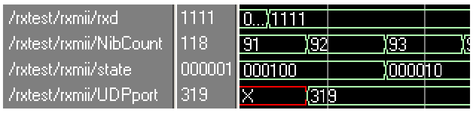
/rxtest/rxmii/rxd 1111
/rxtest/rxmii/NibCount 118
/rxtest/rxmii/state 000001
/rxtest/rxmii/UDPport 319
0... 1111
91 92 93
000100 000010
X 319

Figure 9. The signal of UDPport.

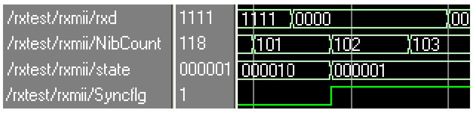

Figure 10. The signal of Syncflg.

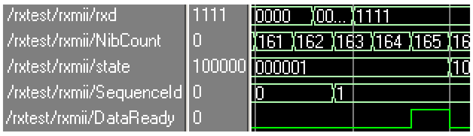

Figure 11. The signals of SequenceId and DataReady.

VI. Conclusion

Standards like IEC61850 could just achieve the accuracy of microsecond level, but IEEE1588 could achieve nanosecond. This paper studied the standard of IEEE1588, and designed a hardware solution using the method of "Cyclone II FPGA + MAC + PHY". The signals of MII interface can be easily extracted. Furthermore, the module can monitor those signals, and identify Sync or Delay_Req messages as the Modelsim simulation shows. All of these will become the cornerstone of the further research of IEEE 1588.

ACKNOWLEDGMENT

The authors would like to thank Shengyun Du and Hongkun Zhao for their help and valuable suggestions.

REFERENCES

[1] U.S.-Canada Power System Outage Task Force: Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations; April 2004.

[2] North American Electric Relaibility Corporation (NERC): Time Stamping of Operational Data Logs; Version 0.995, November 11, 2009.

[3] Fred Steinhauser, Christian Riesch, and Manfred Rudigier, "IEEE 1588 for Time Synchronization of Devices in the Electric Power Industry," 2010 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication.

[4] J. Hu, X. H. Gao, "Application of SNTP-based time synchronization in digital substation," Electric Power Automation Equipment, vol. 29, no. 3, pp. 143-148, Mar. 2009. (in Chinese)

[5] Weibel H, "High precision clock synchronization according to IEEE 1588 implementation and performance issues," Nurenberg: Proceedings of embedded world, pp. 22-24, 2005.

[6] IEEE, "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," IEEE 1588, Jul. 2008.

[7] DAVICOM, DM9000 ISA to Ethernet MAC Controller with Integrated 10/100 PHY Datasheet,Version: DM9000-DS-F03, Jul. 2004.

[8] Intel, LXT971A 3.3V Dual-Speed Fast Ethernet PHY Transceiver Datasheet, Order Number: 249414-002, Aug. 2002.

[9] IEEE, "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications," IEEE Std 802.3 2000 Edition, pp. 496-508, 2000.

[10] Sergio scaglia, "The Embedded Internet TCP/IP Basics, Implementation and Applications," Pearson Education limited, pp. 37-70, 2007.