17th Symposium IMEKO TC 4, 3rd Symposium IMEKO TC 19 and 15th IWADC Workshop
Instrumentation for the ICT Era
Sept. 8-10, 2010, Kosice, Slovakia

Comparison of IEEE 1588 Implementations using the STM32 Connectivity Line processor

Jan Breuer[1], Jan Fischer[1], Jaroslav Roztočil[1]

[1] Department of Measurement, Faculty of Electrical Engineering, Czech Technical University, Technická 2, 166 27 Praha, Czech Republic
{breuejan, fischer, roztocil}@fel.cvut.cz

***Abstract*-**In distributed system containing multiple clocks, precision time synchronization can by achieved using IEEE 1588 protocol. This paper presents a hardware assisted and a software implementation using the STM32 Connectivity Line (STM32C) processor. Accuracy of time synchronization depends on stability of master and slave oscillators, timestamping technique and reference clock adjustment capabilities. Allan deviation of both master and slave are measured to determine oscillator stability. Several tests using a hardware and software timestamping are made. Synchronization performance is also tested using an internal reference clock and using internal general purpose timers/counters.

I. Introduction

Distributed network technologies become more common in measurement and control system applications. Most of the nodes in the network uses its own local clock source or internal oscillator. These clocks all run at slightly different frequency and their frequency drift depending on environmental factors.
IEEE-1588™ is standard for a precision clock synchronization protocol for networked measurement and control systems [1] also known as Precise Time Protocol (PTP). It defines the message based protocol allowing system-wide synchronization in the submicrosecond range. The protocol applies to systems that communicate by local area networks supporting multicast messaging including, but not limited to, Ethernet.
The protocol defines logical topology of the network. Master node periodically sends messages containing current master clock and slaves calibrate itself by these messages (see figure 1). To compensate propagation delay, slave sometimes sends the message to the master and master sends back timestamp of receipt. All these times are necessary to calculate time scale and offset.

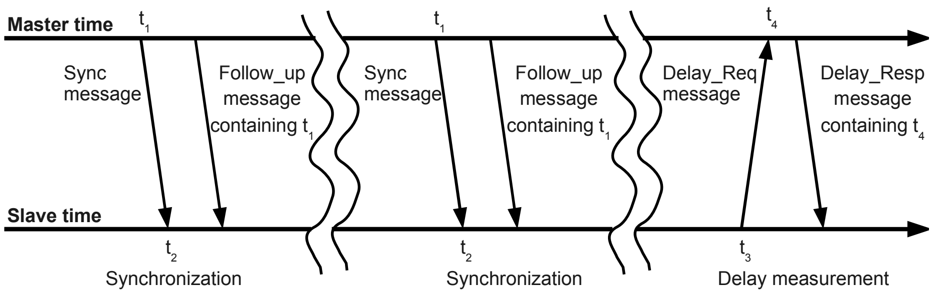

Figure 1. Time synchronization message exchange sequence

II. Protocol Implementation

There are three possible implementations of protocol: software-only implementation [2], hardware-assisted software implementation [3, 4, 5] and hardware-only implementation [6].
In software-only implementation the whole protocol is executed in application level. Accuracy of timestamps in this situation depends on many software and hardware layers and on jitter of each layer. Timestamps implemented at interrupt level could improve accuracy.
Hardware implementations perform timestamping in Media Access Control (MAC) or Physical (PHY) layer (see figure 2). Timestamps on these layers are affected only by jitter of physical layer thus they are more precise than software-only implementation. Hardware-assisted implementation provides precise timestamping of PTP packets, other parts of protocol are implemented in software.

Hardware implementations have mostly its own adjustable reference clock. The reference clock is used for precise timestamping. Application layer adjusts this clock source in fine steps to correct clock offset and frequency according to PTP master.

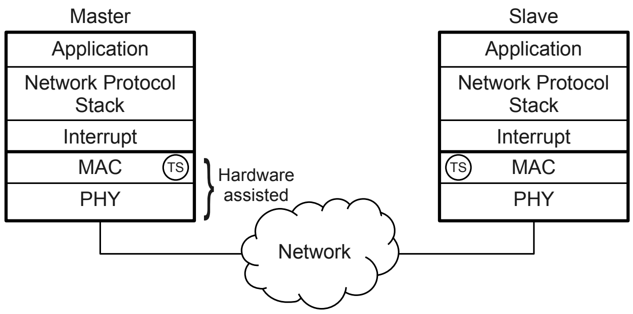

Figure 2. Hardware-assisted timestamping

III. Implementation using STM32 Connectivity Line

A. STM32 Connectivity Line processor overview

STM32 Connectivity Line is a processor line from ST Microelectronics based on the ARM Cortex-M3 core. It contains all common peripherals like GPIO, UART, SPI, I2C, I2S and communication peripherals like USB, CAN and Ethernet MAC.
On chip Ethernet MAC has built-in timestamping support designed for IEEE 1588. It also contains separate reference clock with fine setting of compensation registers. Pulse per second (PPS) could be also propagated to IO pin.

B. Hardware layer

In hardware part the reference clock maintaining actual timestamp in 64 bits format is implemented. The speed of the reference clock updating is fine adjustable allowing digital correction of crystal oscillator's frequency. Both offset (additive component) and frequency updating constant (multiplicative component) of internal clock are adjustable.
The reference clock is incremented by 20 ns increments so theoretical resolution should be also 20 ns. Because of short term stability degradation of crystal oscillator caused by the processor, resolution is at least ten times worse.
The timestamps of incoming and out coming packets are generated by means of the reference clock. This approach guarantees allocation of timestamps for a relevant packet just in time of its arrival/departure, as defined in Specification. The processor also allows to output PPS signal on corresponding pin derived directly from reference clock. The remaining parts of PTP protocol are performed exclusively in software layer.

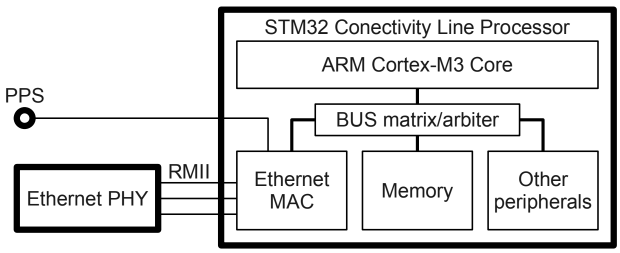

Figure 3. STM32C board block diagram

C. Software layer

The software layer is composed of several mutually interconnected segments. The first segment represents controller of MAC interface, mediating basic requirements of network stack. The MAC interface is set to enable receiving of multicast which is an essential condition for PTP protocol.
Next segment is formed by the network stack, supporting cooperation with User Datagram Protocol (UDP) packets and also with Internet Group Management Protocol (IGMP). As a suitable candidate for this function LwIP stack has been chosen. No further requirements are posed on network stack as all exact timestamping of packets are already performed in MAC circuits.

The last segment (application layer) is an implementation of Precise Time Protocol itself. In this part messages serving for synchronization are received and transmitted. Moreover this segment reads out information on timestamps and based on these data adjusts parameters required for operation of internal clock (i.e. calculation of additive and multiplicative constants).
The implementation of application layer of PTP was taken from Precise Time Protocol Daemon (ptpd) [2]. It implements whole part of PTPv1 protocol stack and clock servo for precise clock adjustment.

D. One-way synchronization

Experiments were made by Agilent E5818A Trigger Box as master clock and STM32C (STM32F107 kit) as slave clock. Frequency stability of PPS of both were measured. Trigger Box stability was measured against atomic clock frequency standard and STM32C stability was measured against Stanford SR620 frequency counter.

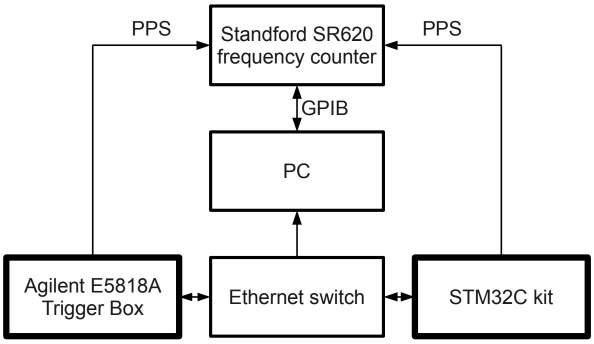

Figure 4. Configuration of devices under test

Three experiments using 3 different implementations were measured. Firstly, hardware timestamping and the internal reference clock were used, secondly software timestamping in interrupt routine and the internal reference clock were used and finally software timestamping and general purpose timers were used. The reference clock is incremented by 20 ns and adjustable by 0.001 ppm steps. General purpose timer is configured to increment by 300 ns and is adjustable by 30 ppm steps.

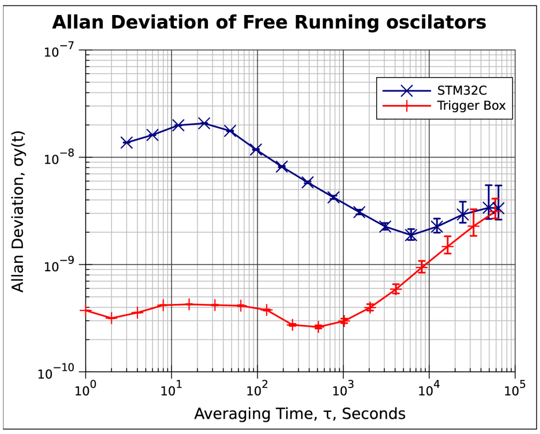

Figure 5. Stability of free running oscillators

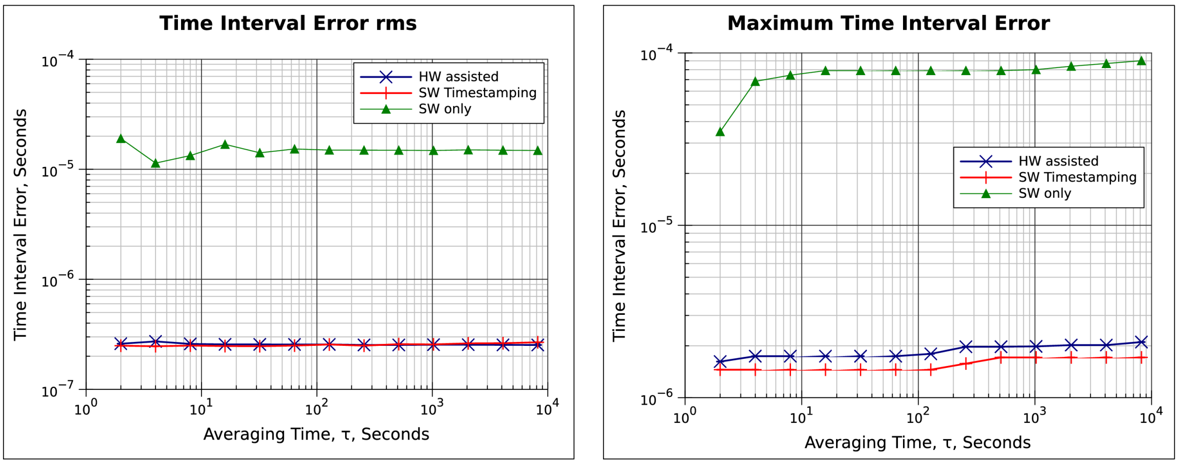

Figure 6. Synchronization stability

As can bee seen software timestamping is slightly better but it shows only stability of synchronization. Mean value of a synchronization offset of hardware timestamping is about 7.8 ns but the offset of software timestamping is about 1,2 µs. Transmit and receive delays must by configured separately and more precisely for software timestamping.

IV. Conclusion

The deviation of the mean value less than one microsecond is achievable when an ordinary crystal oscillator and processor STM32F107 are used. Because of crystal oscillator fluctuations of 8 Hz at 25 MHz base frequency, maximum time interval error could not be better than 2 µs. Because of this degradation difference between software and hardware timestamping is minimal.

Exclusively software solution leads to the increase of deviation of synchronism by one or even two orders. The difference between using reference clock and general purpose timers is larger than the difference using software and hardware timestamping technique.

The advantages of hardware-assisted approach stem from the combination of timestamping on low hardware level and implementation of less time demanding rest of protocol by means of software. Using this approach the uncertainty of synchronization decreased by several orders without necessity to increase of hardware complexity (i.e. no need for application of programmable logic circuits as e.g. FPGA).

References

[1] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)* , vol., no., pp.c1-269, July 24 2008.

[2] Kendall Correll, Nick Barendt, Michael Branicky, *Design Considerations for Software Only Implementations of the IEEE 1588 Precision Time Protocol* , 2005.

[3] *Reference manual, STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM-based 32-bit MCUs*, STMicroelectronics, 2009, pages 880-886, 935-939.

[4] Patrick Ohly, David N. Lombard, Kevin B. Stanton, *Hardware Assisted Precision Time Protocol. Design and case study,* Intel GmbH.

[5] Hardware-Assisted IEEE 1588 Implementation in the Intel® IXP46X Product Line, Intel Corporation, 2005.

[6] Teodor Neagoe, Maher Hamdi, "A Hardware IEEE-1588 Implementation with Processor Frequency Control", *Arrow Resource magazine*; 2006.