

The World Leader in High Performance Signal Processing Solutions



Implementation of IEEE 1588 Precision Time Protocol on Embedded Processors

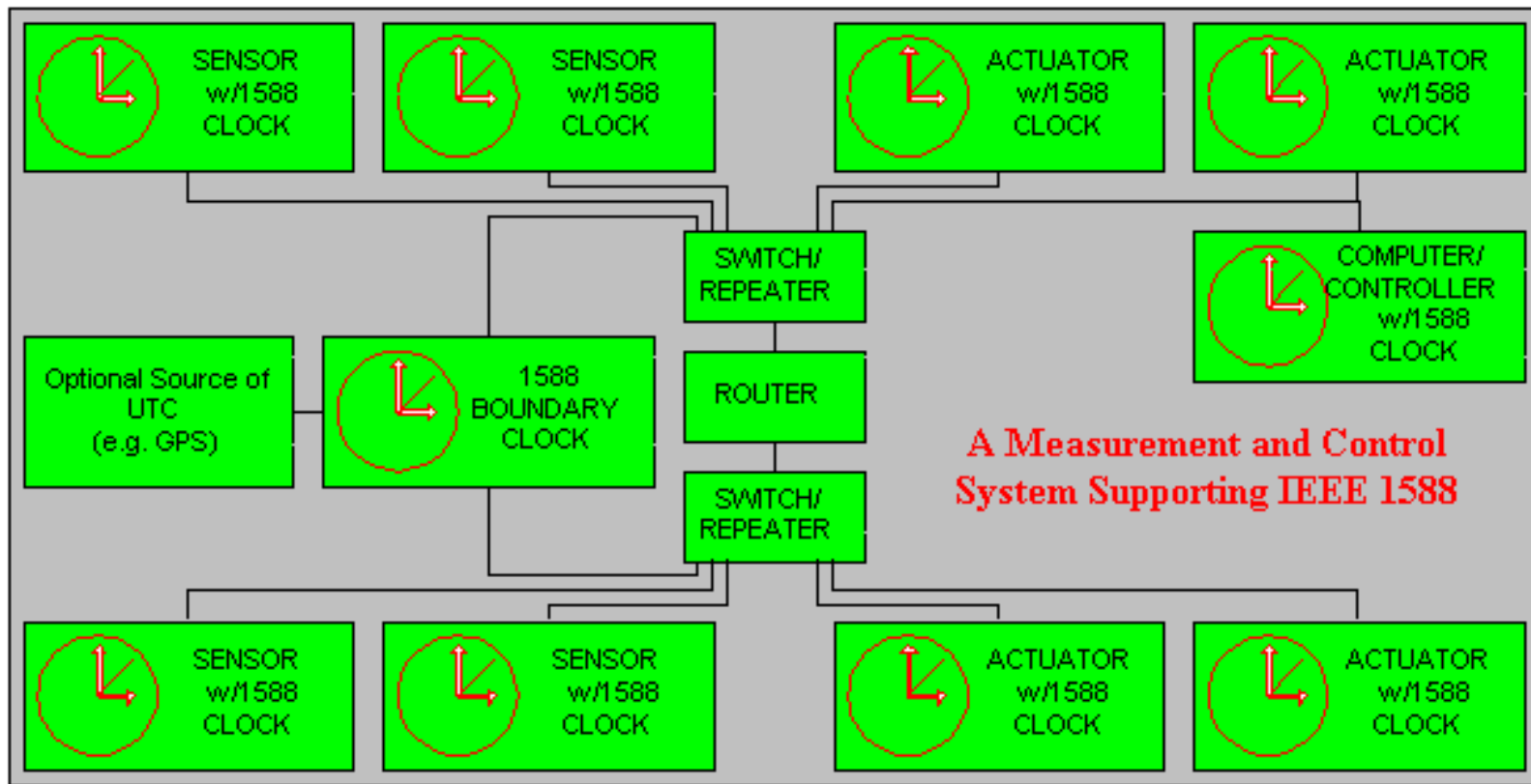
Jiang Wu
Analog Devices Inc.



Agenda

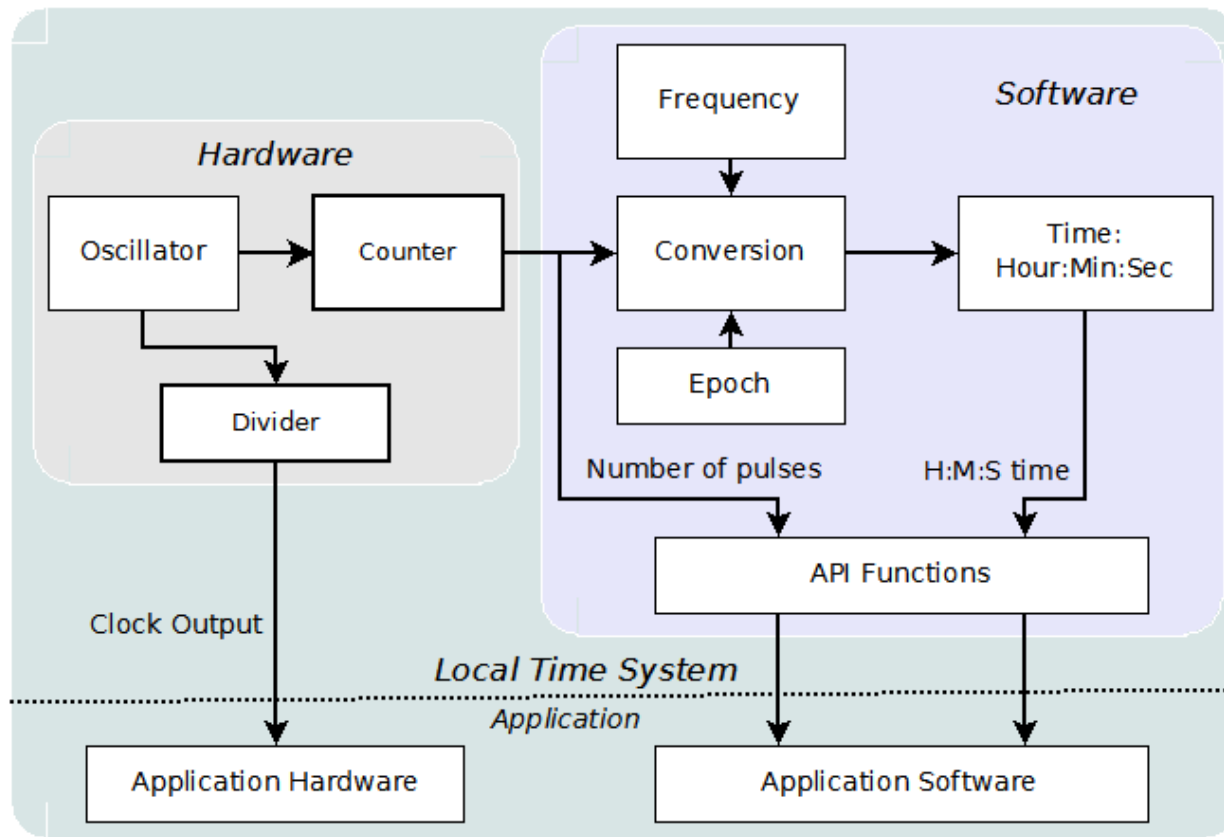
- ◆ **General time representation and synchronization**
- ◆ **Introduction to IEEE 1588**
- ◆ **Integration of IEEE1588 to Embedded Systems**
- ◆ **Example implementation on Blackfin processor**
- ◆ **Demo ...**

Time Synchronization Is Required



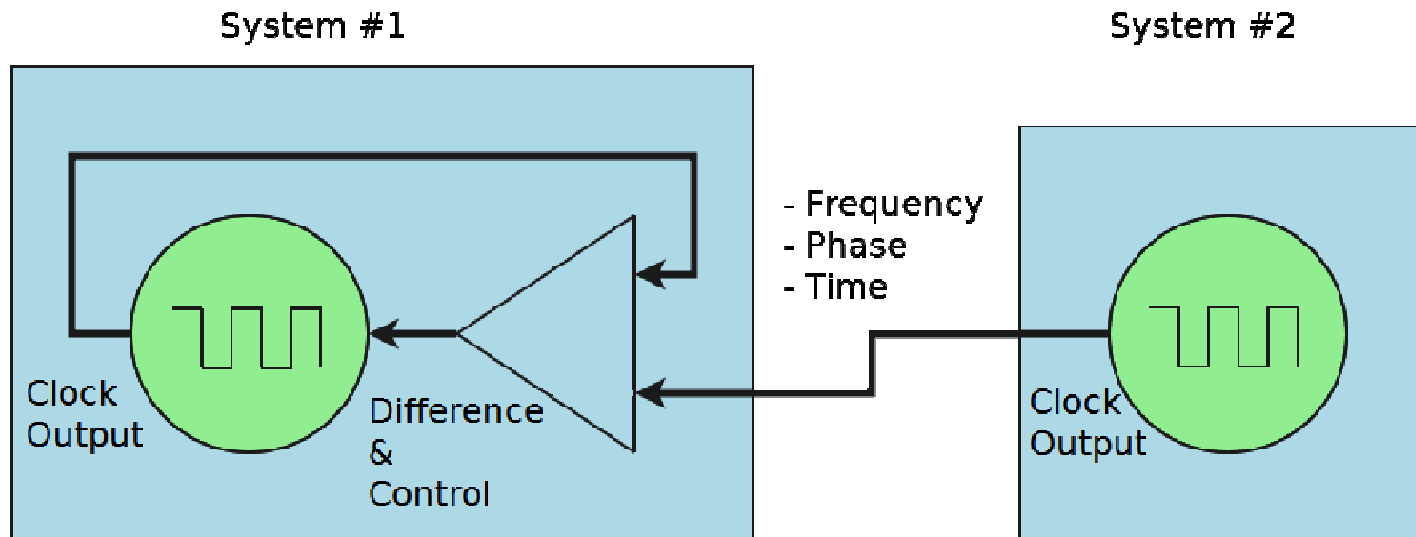
Source: <http://ieee1588.nist.gov/>

Time Representation



- ◆ Time source – oscillator/clock
- ◆ Hardware representation – derived clocks
- ◆ Software representation – number of clocks, h:m:s

Time Synchronization



- ◆ **Feedback control system**
- ◆ **Frequency/phase based - PLL**
- ◆ **Absolute time based – NTP, IEEE 1588**



Time Difference

- ◆ Time differences are measured periodically (discrete)
- ◆ Time differences are accumulated together with frequency difference and clock random error during intervals.

$$e_{n+1} = e_n + R_n(T) + D_n(T)$$

- ◆ Usually the frequency is the one that could be adjusted to minimize the time differences.

Transmission of Time Information

- ◆ Time information is sent from one system to another on a certain path, such as Ethernet, resulting in Path Delay.
- ◆ To obtain the correct time difference, path delay must be taken into account.
- ◆ Round-trip delay is usually measured and symmetric path delay is assumed ($t_{pd1} = t_{pd2}$).

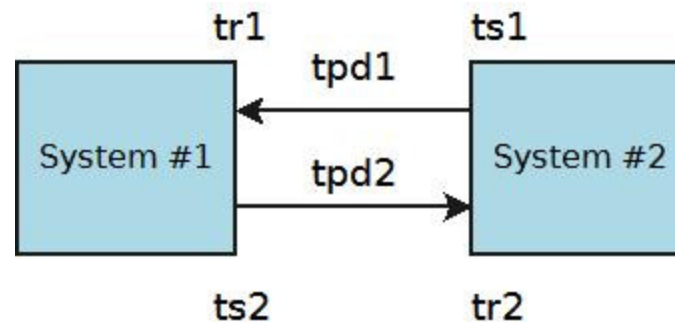
$$t_{diff1} = t_{r1} - (t_{s1} + t_{pd1})$$

$$t_{diff2} = (t_{s2} + t_{pd2}) - t_{r2}$$

$$t_{pd1} = t_{pd2}$$

$$t_{diff1} = t_{diff2} = t_{diff}$$

$$t_{diff} = \frac{(t_{r1} - t_{s1}) + (t_{s2} - t_{r2})}{2}$$





Time Synchronization Options

◆ NTP or IEEE 1588

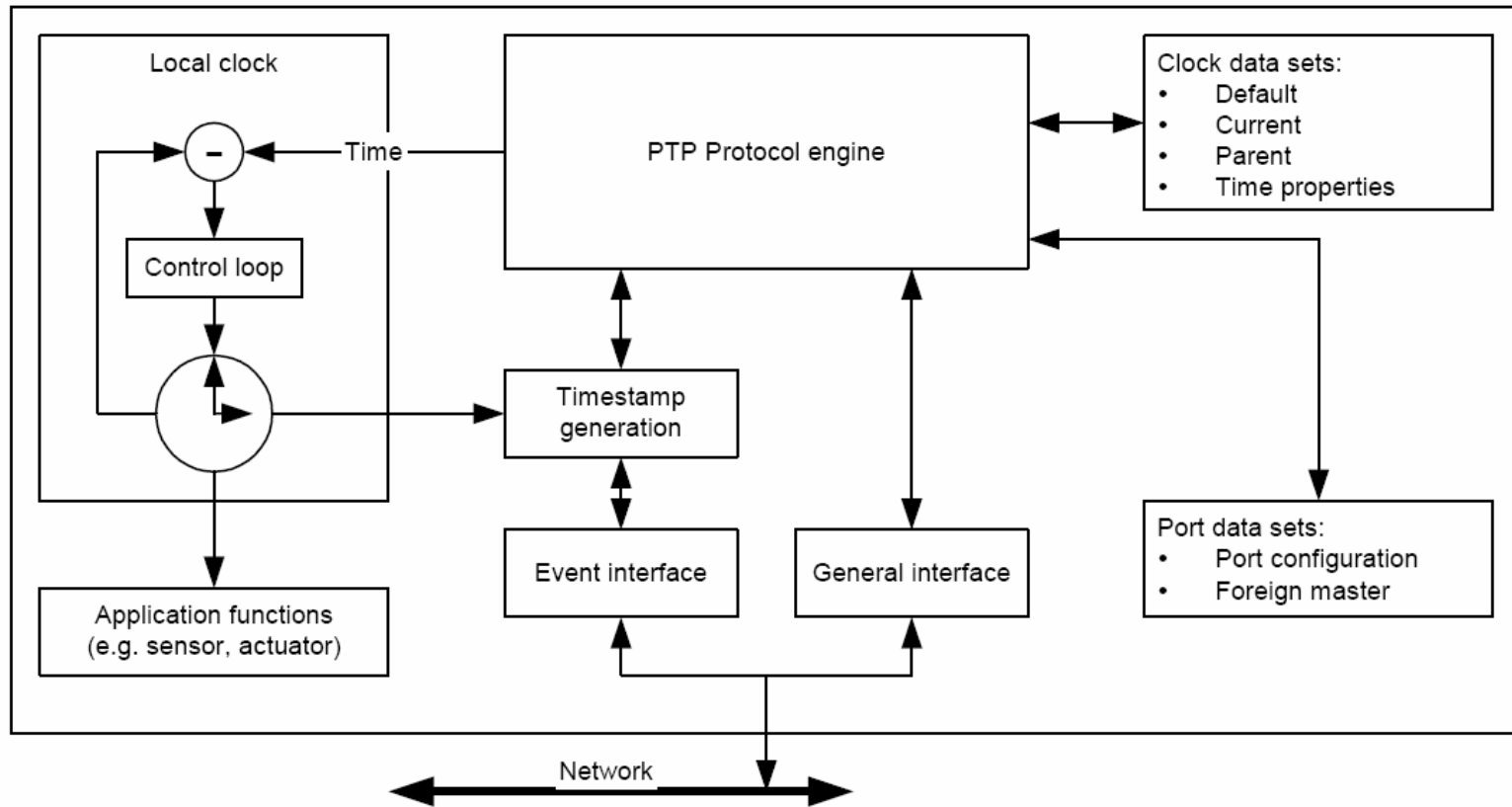
- Both are network based, but have key differences
- Different scope
 - ◆ NTP target is nodes widely dispersed on Internet.
 - ◆ IEEE 1588 target is groups of locally networked (a few subnets) nodes.
- Different level of accuracy
 - ◆ NTP target is milli-seconds over public Internet, and hundreds micro-seconds over local area network
 - ◆ IEEE 1588 target accuracy is sub-micro/nano-seconds
- Different administration
 - ◆ NTP needs configuration
 - ◆ IEEE 1588 is self-organized



IEEE 1588 PTP Overview

- ◆ One particular implementation of time synchronization (time-based), capable of nanosecond level accuracy
- ◆ Defines a mechanism for exchanging time information (messages)
- ◆ Needs the general “symmetric path delay” assumption
- ◆ Independent of communication technology, such as 802.3, DeviceNet etc.
- ◆ Automatically determines Master and Slave clocks
 - Master clock – assumed to have the best quality
 - Slave clock – adjusts its own clock to follow the Master
- ◆ Two versions so far – 2002 and 2008
 - Same mechanism
 - More message types for peer-to-peer delay

Components of IEEE 1588



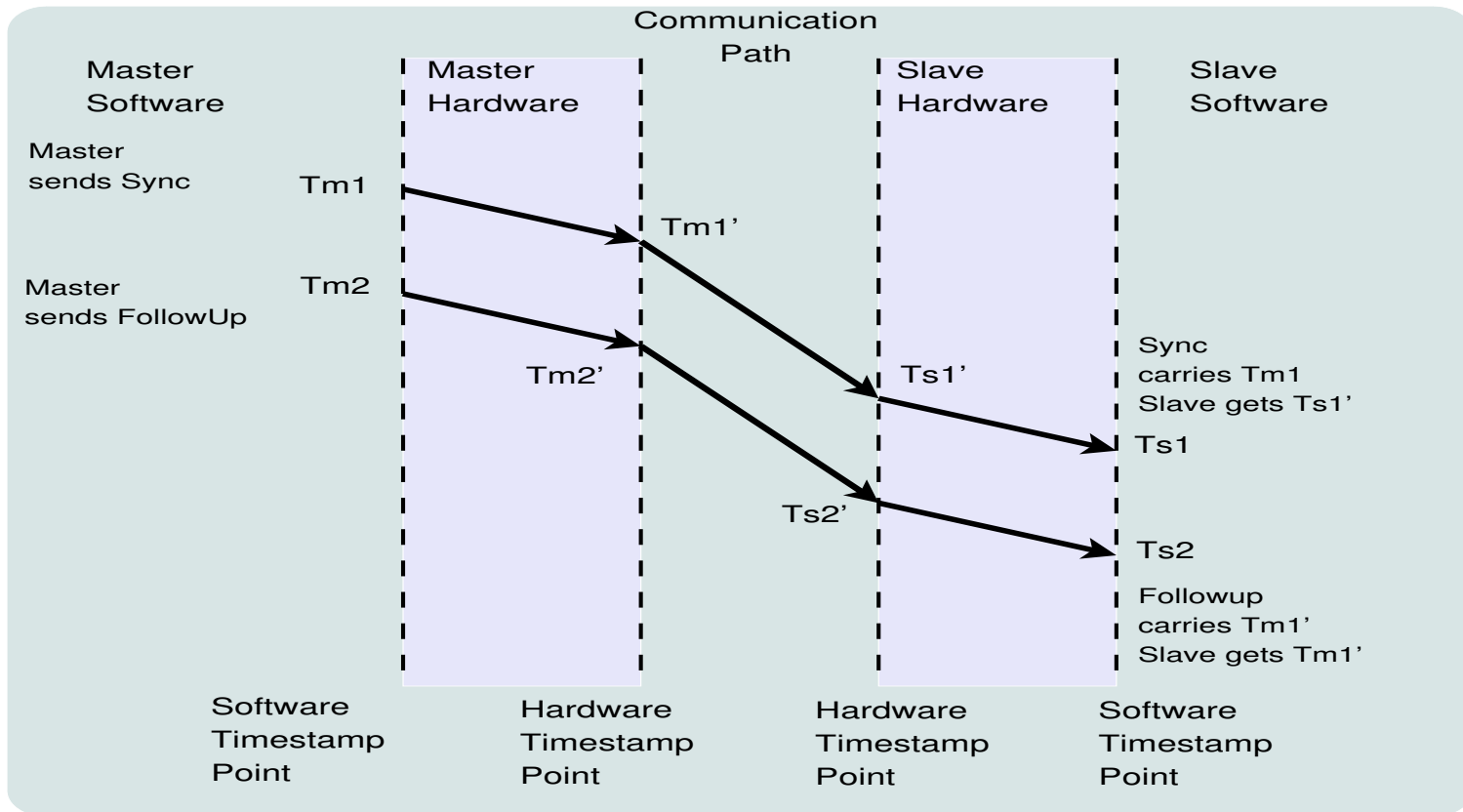
Source: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems



PTP Message

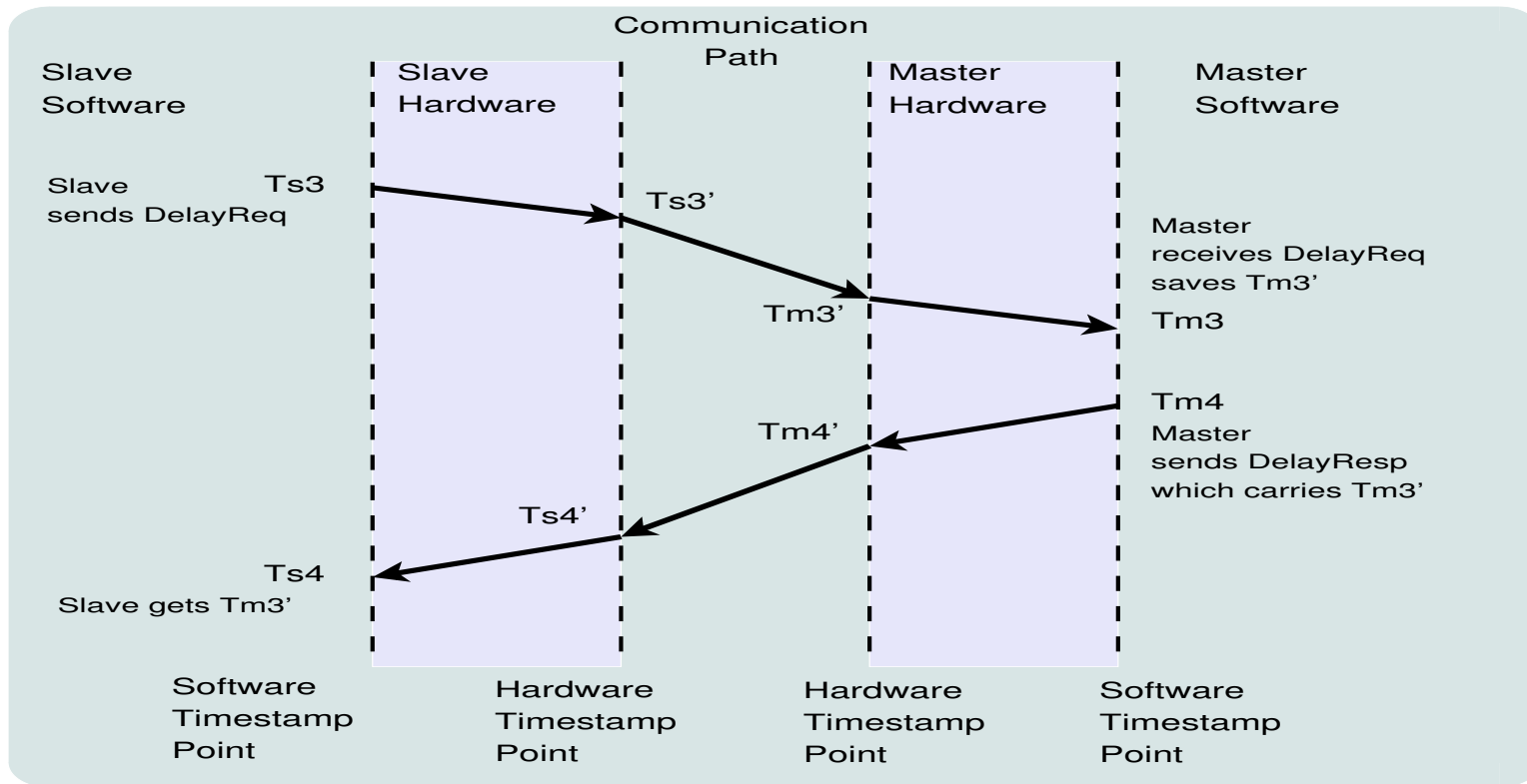
- ◆ **Event messages (timestamped)**
 - Sync
 - Delay_Req
 - Pdelay_Req
 - Pdelay_Resp
- ◆ **General messages (not timestamped)**
 - Follow_Up
 - Delay_Resp
 - Pdelay_Resp_Follow_Up
 - Announce
 - Management

Sync and FollowUp Message



$$T_{msd} = (T_{s1}' + T_{ms}) - T_{m1}'$$

DelayReq and DelayResp Messages



$$T_{smd} = T_{m3'} - (T_{s3'} + T_{ms})$$



Calculation of Path Delay and Time Error

$$T_{msd} = (T_{s1'} + T_{ms}) - T_{m1'}$$

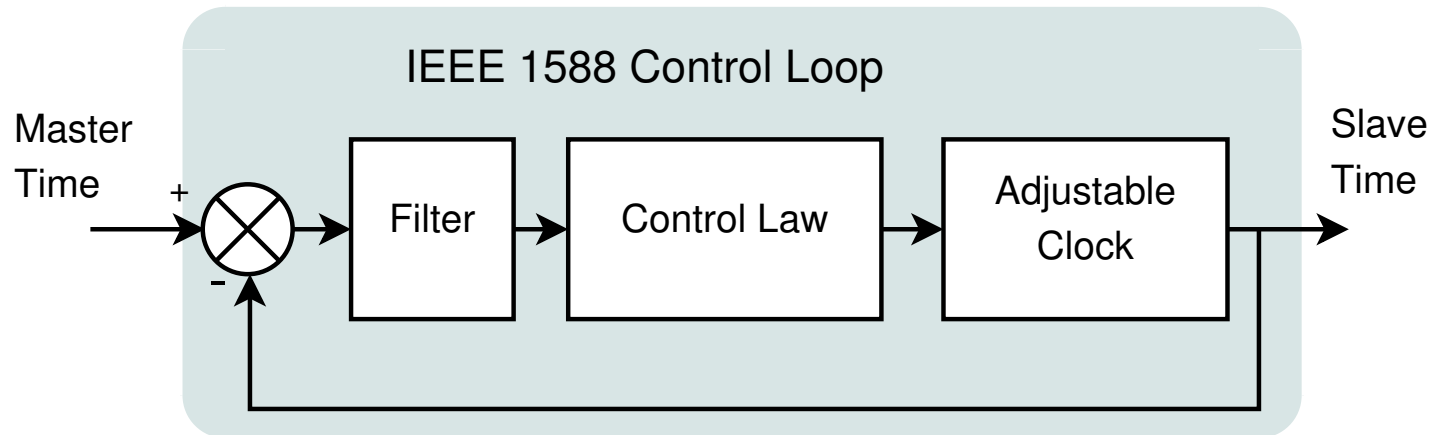
$$T_{smd} = T_{m3'} - (T_{s3'} + T_{ms})$$

$$T_{msd} = T_{smd} = T_d$$

$$T_d = \frac{1}{2} [(T_{s1'} - T_{m1'}) + (T_{m3'} - T_{s3'})]$$

$$T_{ms} = T_d - (T_{s1'} - T_{m1'})$$

Control Loop



- ◆ Time errors are usually filtered to remove the noise of measured path delay
 - Moving average.
- ◆ Control law determines the adjustment of clock,
 - PID controller is most commonly used
- ◆ Filter and control law are not specified by the standard, instead are implementation specific.



Best Master Clock

- ◆ **Every participating clock**
 - Broadcasts its own properties in Announce messages.
 - Collects the properties of all clocks,
 - Runs the same Best Master Clock algorithm by comparing the properties,
 - In comparing the properties, applies tie-breaking rule (clock identification),
 - Concludes with the same result about who is to be Master Clock
- ◆ **It is self-organized and no negotiation is necessary.**
- ◆ **Different from other time synchronization mechanism, such as NTP.**



Factors Impacting Synchronization Accuracy

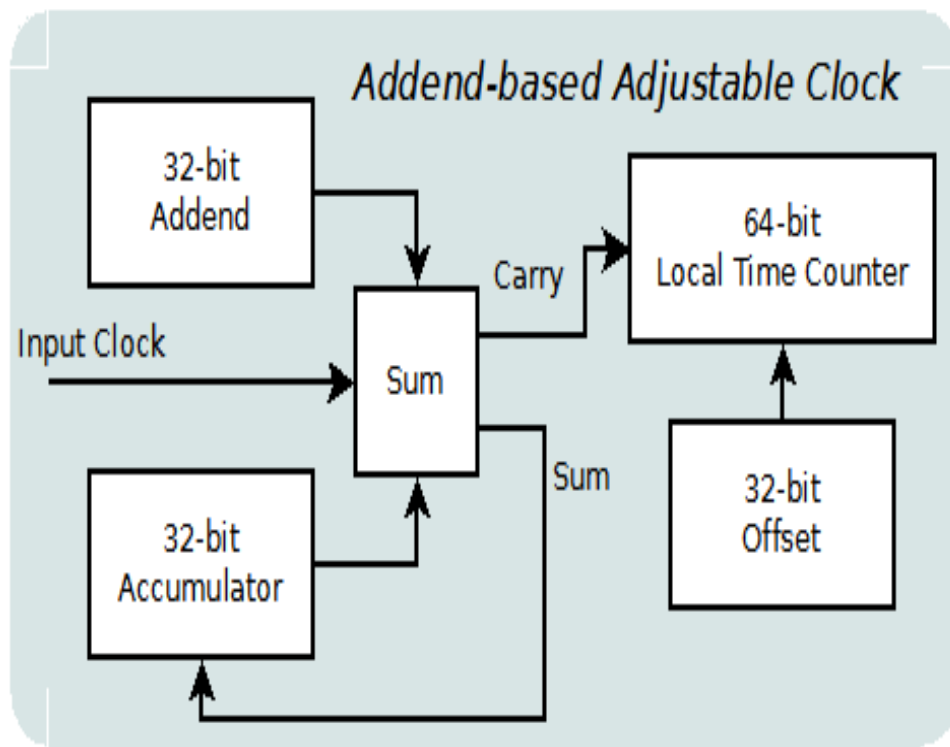
- ◆ **Path delay**
 - Symmetry – delay measurement assumption
 - Variation – measurement noise
- ◆ **Drift and jitter characteristics of clocks**
 - Master clock – reference input change
 - Slave clock – control object change
- ◆ **Control law**
- ◆ **Resolution of the clocks**
 - Quantization error of the measurement
- ◆ **Sync messages interval**
 - Control period
- ◆ **Delay measurement interval**
 - Measurement noise



Integrate IEEE 1588 to Embedded Systems

- ◆ **A communication channel for message exchange. Currently Ethernet is the most common one, and discussed here.**
- ◆ **A clock whose frequency could be adjusted dynamically.**
- ◆ **IEEE 1588 messages detection.**
- ◆ **Timestamping capability – for better performance, it should be close to the physical cable to satisfy the “symmetry” requirement.**
- ◆ **Message parsing, path delay calculating, filtering, controlling, state machine maintaining.**
- ◆ **Extra features allowing easy integration into applications**

Adjustable Clock Example - Addend-based Clock



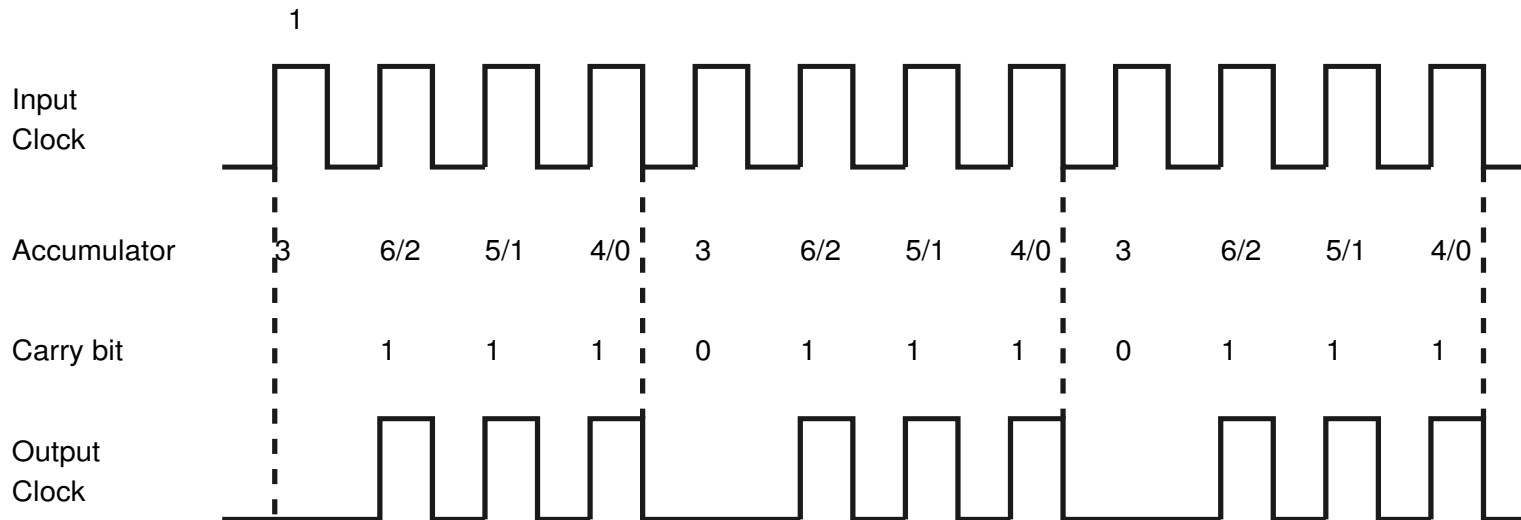
- ◆ For every input clock (F_{in}), Accumulator is incremented by Addend. The carry bit of the addition is used as the output clock (F_{out}).

$$F_{out} = F_{in} \cdot \frac{A}{2^{32}}$$

- ◆ The width of the Accumulator determines the clock adjustment resolution.

Addend-based Clock

- ◆ Purely digital clock, easy to implement in digital processors
- ◆ Direct clock output has large jitter



◆ Example:

- Accumulator is 2-bit wide
- Addend is 3
- Average output frequency is

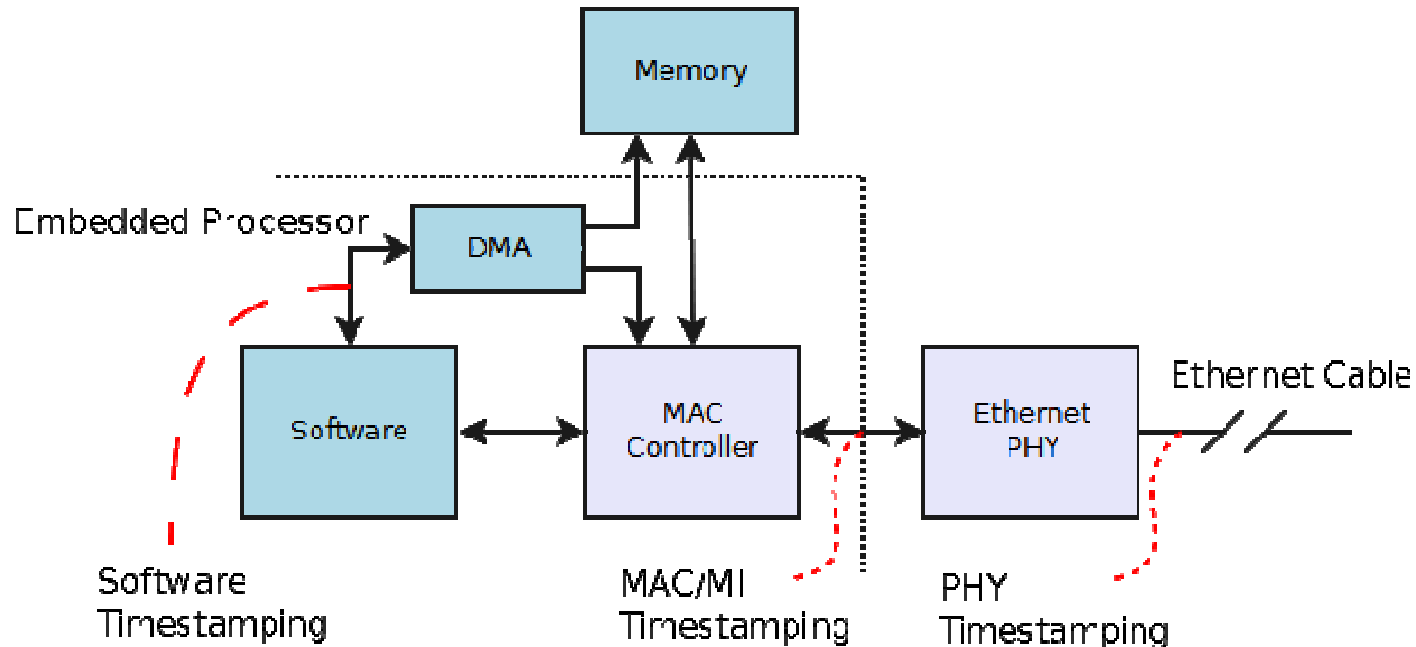
$$F_{out} = F_{in} \cdot \frac{3}{2^2} = \frac{3}{4} F_{in}$$



Packet Detection

- ◆ Using Ethernet, all IEEE 1588 messages are sent/received as Ethernet packets (one message per packet).
- ◆ Event messages/packets need to be distinguished from other regular Ethernet packets
 - On MAC layer, EtherType (0x88f7) and destination MAC address(01-1B-19-00-00-00, 01-80-C2-00-00-0E) can be used.
 - On UDP layer, port number (port 319, 320) and destination IP address (224.0.1.129, 224.0.0.107) can be used.
 - Other fields in message header can also be used to verify the message, such as PTP version, message type, etc.
- ◆ An array of (*offset, value*) pairs can be used to specify which fields match with their corresponding values.

Timestamping Points



- ◆ There are three possible timestamping points.
- ◆ Each has different characteristics of Symmetry and Variation.
- ◆ Accuracy:
 - PHY – nano-second accuracy
 - MAC/MI – nano-second accuracy, depending on the jitter of the PHY delay (<1ns)
 - Software – usually micro-seconds accuracy depending on the software latency



Retrieving Timestamps

- ◆ **The timestamps must be accessible to software**
- ◆ **There are two options of how to read timestamps with regard to message read**
 - **Asynchronous**
 - **Synchronous**



Asynchronous Timestamp Read

- ◆ **PHY timestamps – PHY internal registers, MII management channel**
- ◆ **MAC timestamps – MAC internal registers, MAC to core interface, such system registers.**
- ◆ **Software reads the timestamp registers asynchronously to getting the corresponding messages through ISR**
 - **A timestamp FIFO to compensate for the potential latency of software's timestamp read**
 - **FIFO depth depends how often event messages could arrive and the maximum software latency**
 - **Message ID is usually used to match messages with timestamps, since the message could possibly be lost on their way to software.**
 - **Message ID field of messages should be captured together with timestamps**



Synchronous Timestamp Read

- ◆ **Timestamps are inserted in messages (in-band)**
- ◆ **Messages are always delivered to software together with their timestamps.**
- ◆ **Pros**
 - Messages and timestamps always match
 - No FIFO is needed
 - No timestamp interrupt is needed
- ◆ **Cons**
 - Need extra functionality of timestamping unit – reconstruction of packets, CRC recomputed



Extra Features

- ◆ **Clock source options (to drive adjustable PTP clock)**
 - **Processor internal clock**
 - ◆ Most convenient to use
 - **External clock**
 - ◆ Meet specific requirement of clock characteristics of some applications
 - **Network clock**
 - ◆ Use the clock on the Ethernet as the input clock



Extra Features

- ◆ **PPS output**
 - Pulse-per-second physical signal of the PTP adjusted clock
 - Occurs at every second transition
 - Another way to convey time information
 - Can be used for measuring synchronization performance
 - Triggers processor internal interrupt



Extra Features

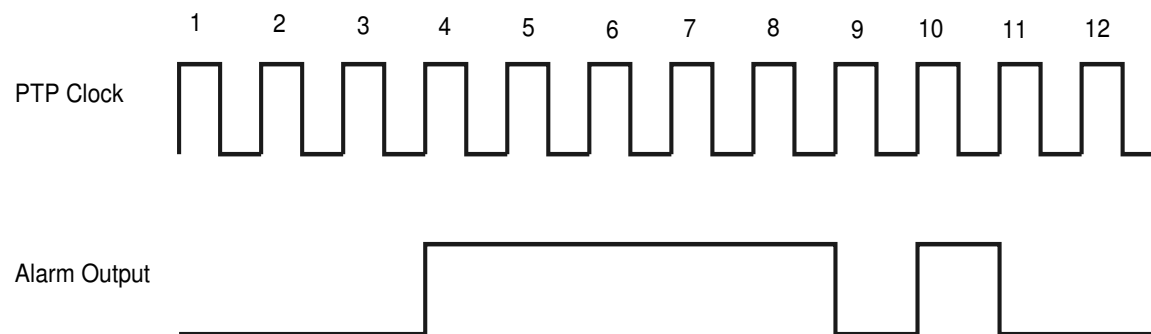
◆ Clock Output

- A physical signal representing the PTP adjustable clock
- Direct clock output or jitter-filtered clock output
- Duty cycle option
- Divider

Extra Features

◆ Alarm

- Triggered at a pre-defined time
- Generates internal interrupt or external output
- Combination of multiple alarm output
 - ◆ Generate arbitrary waveform based on PTP time



◆ Combined alarm output

- Alarm #1: 4
- Alarm #2: 9
- Alarm #3: 10
- Alarm #4: 11



Extra Features

- ◆ **Auxiliary Snapshot**
 - Detects external events (toggling of a signal) and captures timestamps
 - Triggers processor internal interrupt



Software Requirement

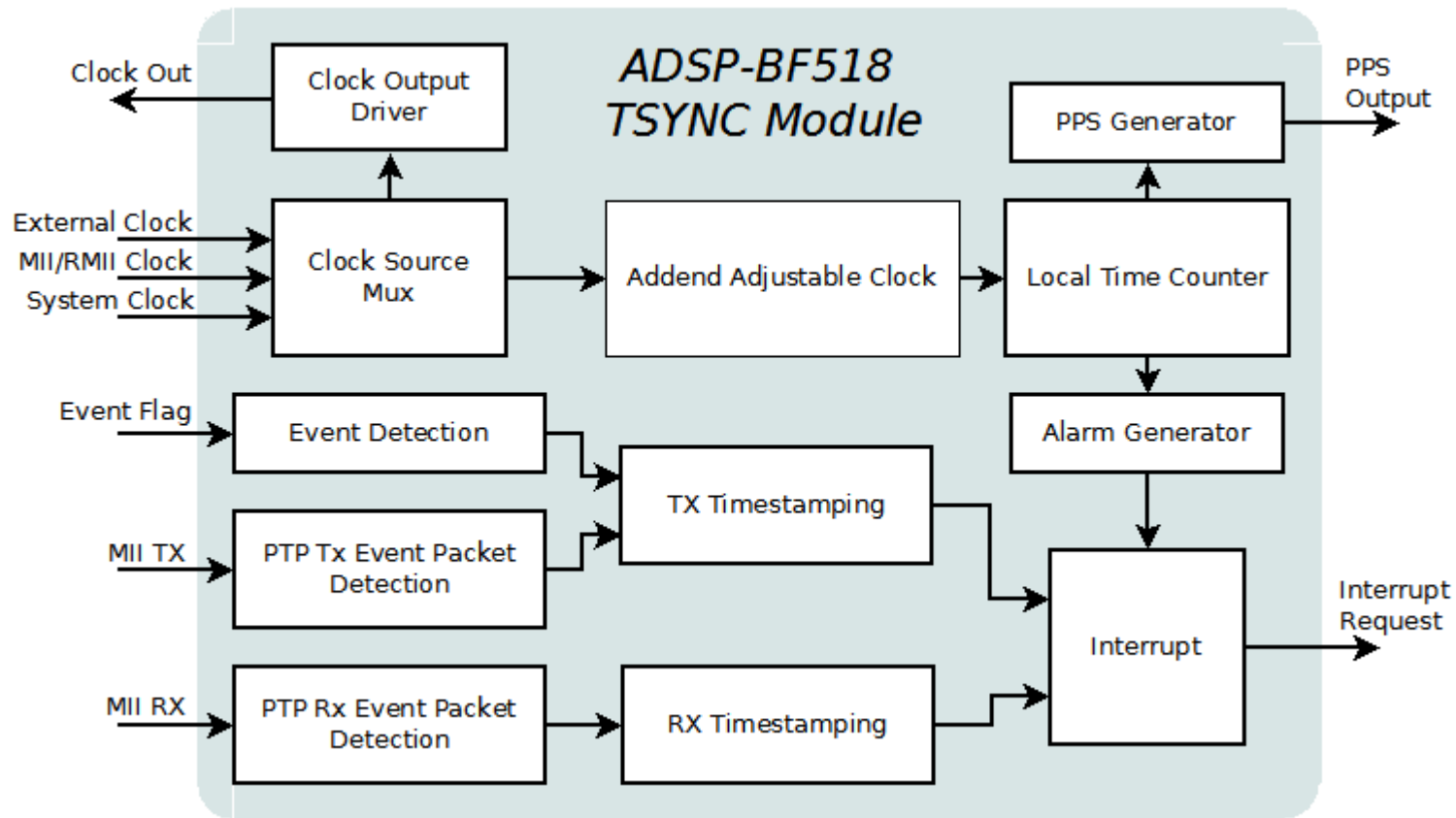
- ◆ **Hardware drivers for adjustable clock control, message timestamping and message delivery**
- ◆ **Software stack for parsing/generating messages, computing and filtering measurements, executing control law, and maintaining state machine**
 - **Independent of the underlying hardware**
 - **Available both commercially (IXXAT Inc.) or as open-source (ptpd)**



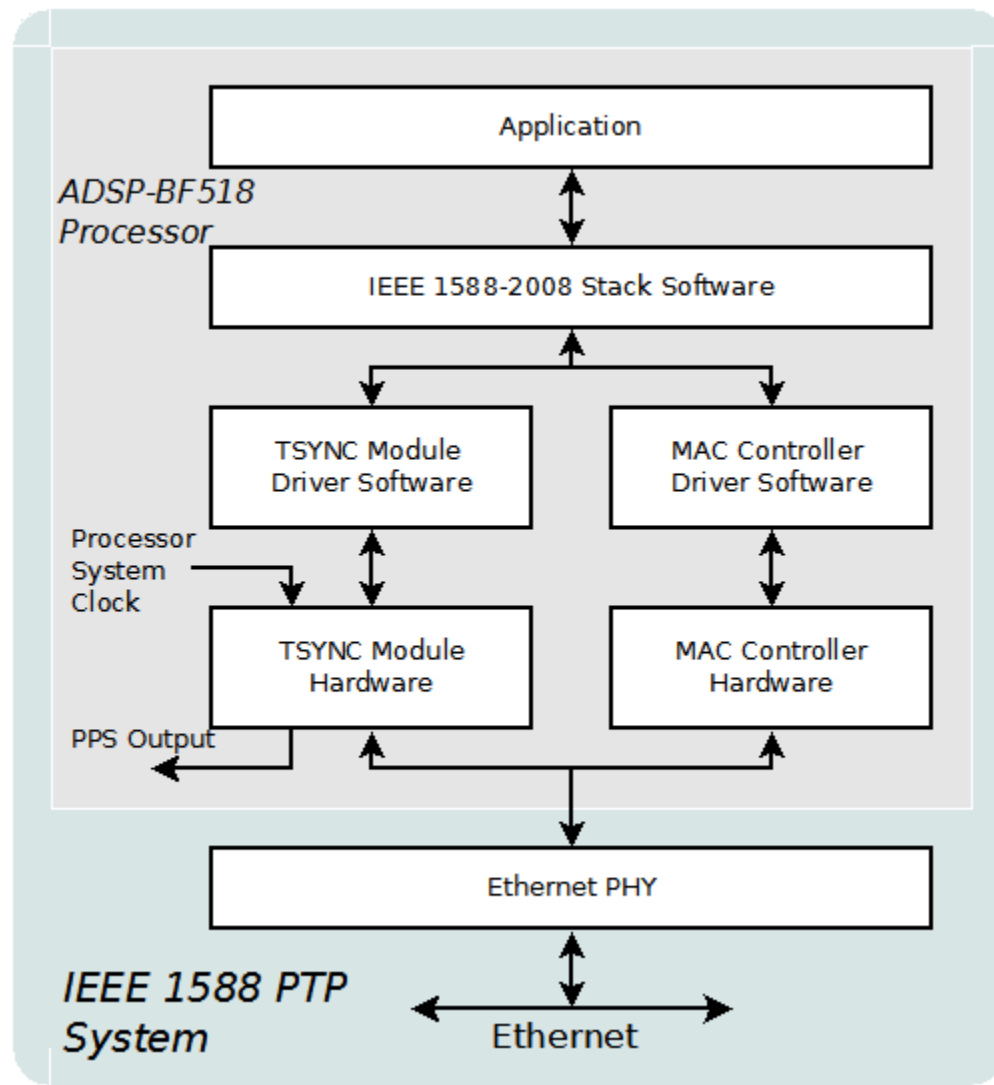
Example Implementation – ADSP-BF518

- ◆ **Built-in Ethernet MAC controller**
- ◆ **Timestamping point at MAC/MII interface**
- ◆ **Addend-based adjustable clock**
- ◆ **Asynchronous timestamp read (system registers)**
- ◆ **Programmable packet detection, supports both versions and both layers (MAC and UDP)**
- ◆ **Extra features**
 - **Clock source options for the adjustable clock**
 - **Direct clock output with divider**
 - **Programmable PPS output (start time and period)**
 - **One alarm with interrupt**
 - **One auxiliary snapshot**

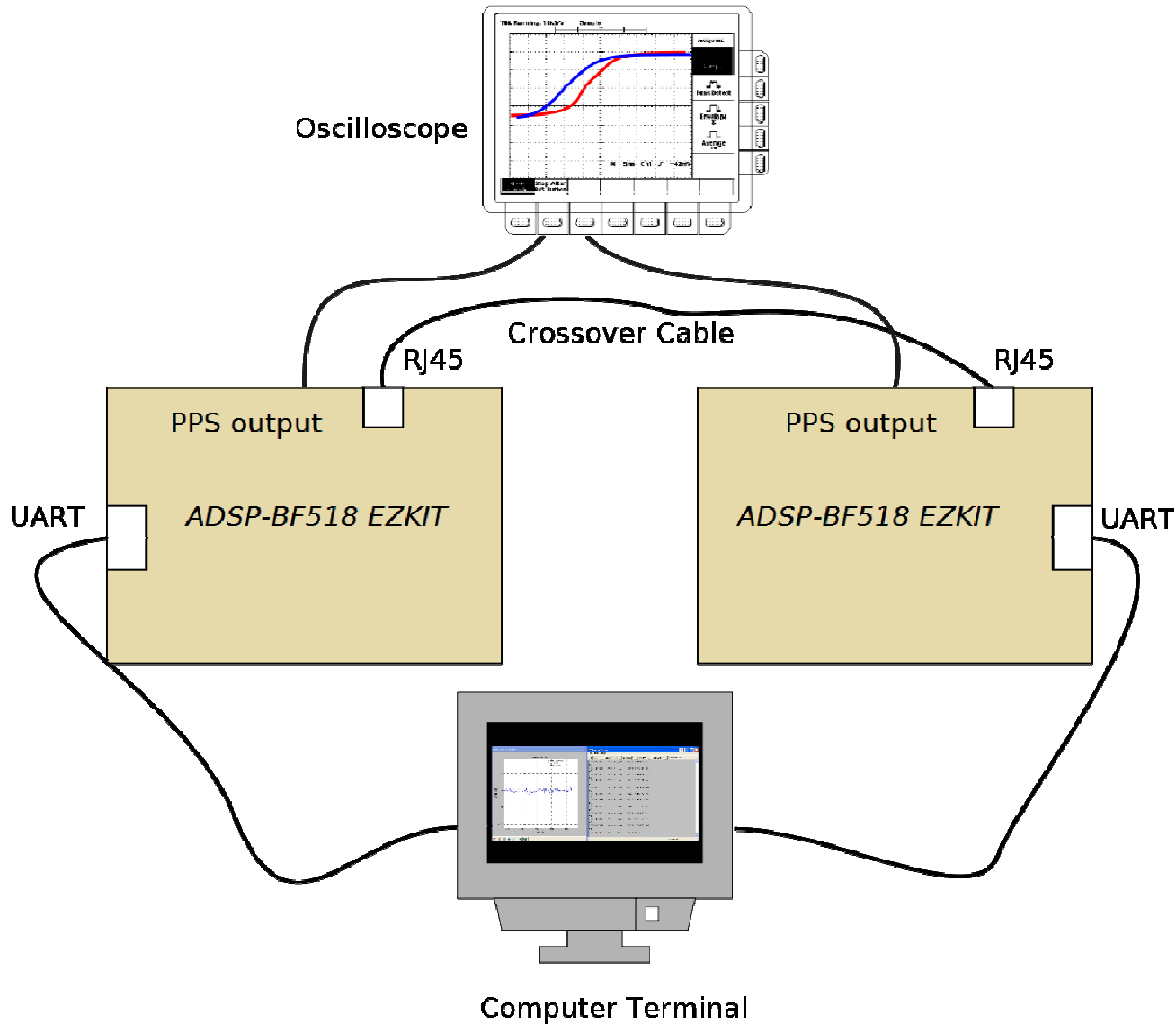
ADSP-BF518 PTP Module



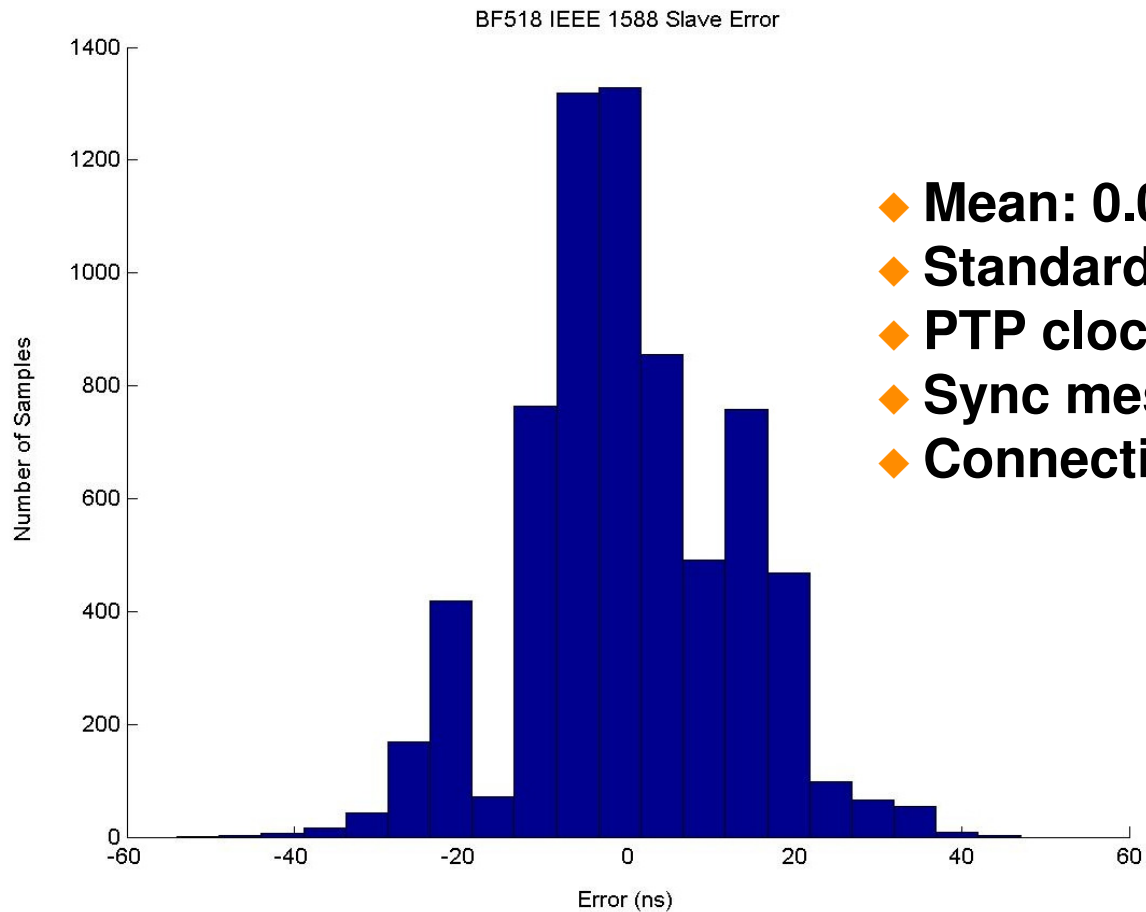
ADSP-BF518 IEEE 1588 Demo Platform



ADSP-BF518 IEEE 1588 Demo Platform



ADSP-BF518 IEEE 1588 Demo Platform



- ◆ Mean: 0.015ns
- ◆ Standard Deviation: 12.96ns
- ◆ PTP clock: 75MHz
- ◆ Sync message interval: 1/4 second
- ◆ Connection: back-to-back



Acknowledge

- ◆ **Analog Devices Inc.**
- ◆ **Rick Gentile**
- ◆ **Robert Peloquin**
- ◆ **Matthew Borto**
- ◆ **All my colleagues**



Questions

◆ Thank you