logic fruit Technologies

Whitepaper

Beginner's Guide to

PRECISION TIME PROTOCOL

By:

Mohammad Rafi

BU Head - Logic Fruit Technologies

What We Learn

1. Key concepts of time and clock
2. Introduction to PTP and its working principle
3. Types of PTP clocks
4. Must know concepts before implementing PTP
5. Bird-eye view of PTP implementation
6. Basics of PTP testing

Overview

In today's world, we are using a large number of connected devices that provide us with a seamless experience in voice calling, data transfer, and many more applications. This creates a critical need for the devices and the systems handling these applications at the backend stay synchronized with each other. With the demand for higher data rates, the synchronization needs have moved from milli-seconds to nano-seconds range. Time synchronization using PTP over the packet network addresses this in a smart and scalable approach.

Concept of Time, Clock, and Time of Day (ToD)

What is time?

Time is the continued sequence of existence and events that occur in an irreversible succession from the past, through the present, and into the future

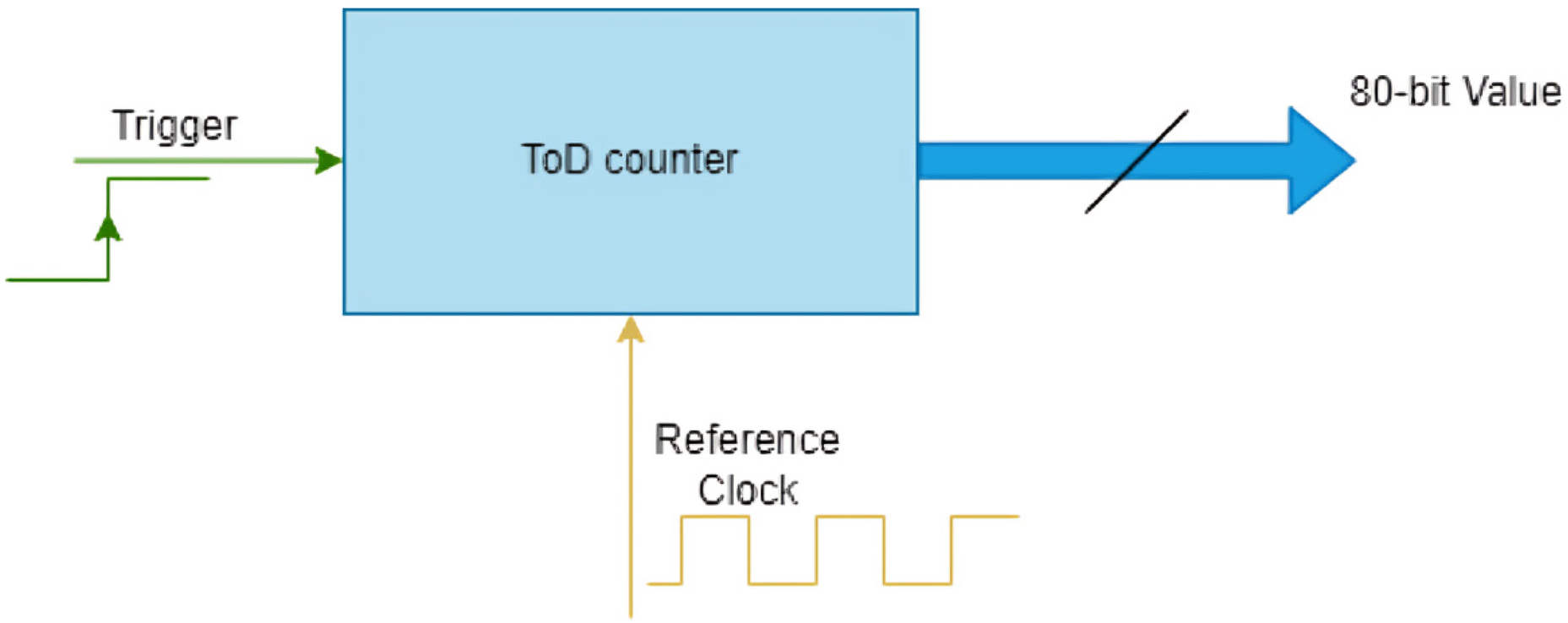
A clock is an entity that maintains time. To maintain time, the clock must run at a fixed frequency (to increment the time at a uniform rate). The clock will also have a phase that indicates the instance at which the transition of time unit (seconds/milliseconds/ nanoseconds etc) happens.

When two clocks are synchronized, they essentially have the same frequency and phase. They are running at the same speed and the transition of a second is happening at the same instance

The clock is an entity that maintains time

ToD – a global counter

IEEE 1588 uses the Wall Clock concept, that is it uses the Time of Day (ToD), not only hours, minutes, and seconds, but also the number of nanoseconds within the second. More precisely, PTP carries time that consists of 48 bits worth of seconds, and 32 bits worth of nanoseconds. The 48-bit seconds is the number of seconds that have elapsed since January 1, 1970, at midnight, which is known as the PTP epoch.



The reference clock decides the resolution of the time measurement. A 125MHz clock can provide 8ns resolution. A 500MHz clock can provide 2ns resolution. To achieve better resolution (and thereby better synchronization) a higher clock frequency is required.

1PPS: 1 Puplse Per Second

In the context of synchronization, 1PPS is an output generated by the ToD counter which has a time period of 1 second. The rising edge of the 1PPS signal represents an instance at which the (binary) bit corresponding to 1-second in the ToD counter gets toggled.

The rising edge of the 1PPS signal is used to verify the phase synchronization of two clocks in the PTP network.

What is PTP

PTP stands for Precision Time Protocol.
A protocol designed to synchronize clocks in a network by exchanging certain messages.

Why do we need it?

In a range of applications especially telecom, synchronization plays an important role. As a use case, we can look at synchronization requirements in 5G.

The precise timing and synchronization requirements for 5G networks are driven by the need to achieve faster speeds, lower latency, and increased cell site densification. Timing and synchronization help prevent messages from interfering with one another and enable smooth cell-to-cell transfers. The shift to packet-based transport and time division duplex (TDD) technologies also requires a higher degree of precision.

Duplexing, multiplexing, and packet-based strategies all rely heavily on timing references to coordinate data transmission, prevent interference, reduce error rates, and compensate for any frequency or phase shifts.

Without timing and synchronization, efficient spectrum utilization and the delivery of high-speed, high-bandwidth wireless services would not be possible.

5G RAN with centralized units (CUs) and distributed units (DUs) fulfills the functionality of Base Band Units and needs tightly controlled absolute and relative timing between these elements to meet the 5G use case requirements.

Synchronization of the fronthaul network allows the RAN to function properly, regardless of the distance between CU and DU.

When 5G network nodes are out of sync, received signals cannot be demodulated properly. This can lead to a high bit error rate, delay, and jitter that compromise the customer experience.

GNSS Signal Quality acquired by the satellite antenna must be highly reliable to meet the requirements of 5G. Precise Timing and Synchronization requirements of 5G cell sites have made even small-time variations unacceptable. While 3G and 4G networks only need one satellite line of site for synchronization, 5G networks require a lock on 4 or more satellite positions to minimize the impact of satellite location. Having a high-precision GNSS receiver system also adds to the cost of installation and maintenance.

PTP Port Roles:

A physical or logical interface with a network stack to transmit or receive the PTP packets is considered a PTP port. A PTP port can take one of the two roles as described below.

- **Master:** The port through which a clock distributes time to slave ports.
- **Slave:** The port through which a clock synchronizes itself to a master.

PTP Clock Types:

There are different kinds of Clocks defined in PTP IEEE 1588. As per the ITU T standards, these clocks can have different roles and names. The clocks as defined by IEEE1588 are as follows:

1. Ordinary Clock(OC)

An Ordinary Clock has only one port for the transmission and reception of the PTP packets. An ordinary clock can be a GM to distribute time or a slave to get the time from a GM.

2. Boundary Clock (BC)

A Boundary clock has a minimum of two PTP ports. One port must be a Slave port that synchronizes with an upstream Master clock. The rest of the ports are master ports that distribute time to the downstream slave clocks. BC can also act as GM with all of its ports as Master ports when no GM is available or when it is chosen as GM by all other clocks through BMCA.

3. Transparent Clock (TC)

A Transparent Clock has at least two PTP ports. It assists in the delay measurement between a Master and a Slave by including a Correction Factor (CF) that tells the Slave how much delay it has added to the message passing through it. There are two types of TCs based on the Delay mechanism, e2e-TC, and p2p-TC.

PTP Delay Mechanism

Two Types of PTP Delay Mechanism

- ☞ End to End delay mechanism
- ☞ Peer to Peer delay mechanism

1. End to End (Delay Request/Delay Response):

In this type of delay mechanism, the Slave measures the delay between itself and the Master (thus End-to-End). The Master and Slave send PTP messages called ***DELAY REQUEST*** and ***DELAY RESPONSE*** back and forth between the two, allowing the delay to be measured. Once the delay is known, the Slave can adjust its clock to achieve time synchronization (Frequency and Phase) with the Master.

2. Peer to Peer:

In this type of delay mechanism, each PTP node measures the delay between its port and the port of the immediate peer (PTP node) in the network. As the Master sends its time using SYNC messages towards Slave(s), each network element along the way receives the SYNC message and adds a correction to the SYNC message. The correction includes the measured wire delay of the input port where the SYNC message was received.

How it works – Basics

For a Slave clock to synchronize its time with a master clock, it needs to know two values

- ☞ The time in the Master clock
- ☞ The path delay (time taken for the message to travel from master to slave) between Master and Slave. Once the above two values are a vailable at the slave, the slave can calculate the difference between the time running in its clock and the time in the master clock

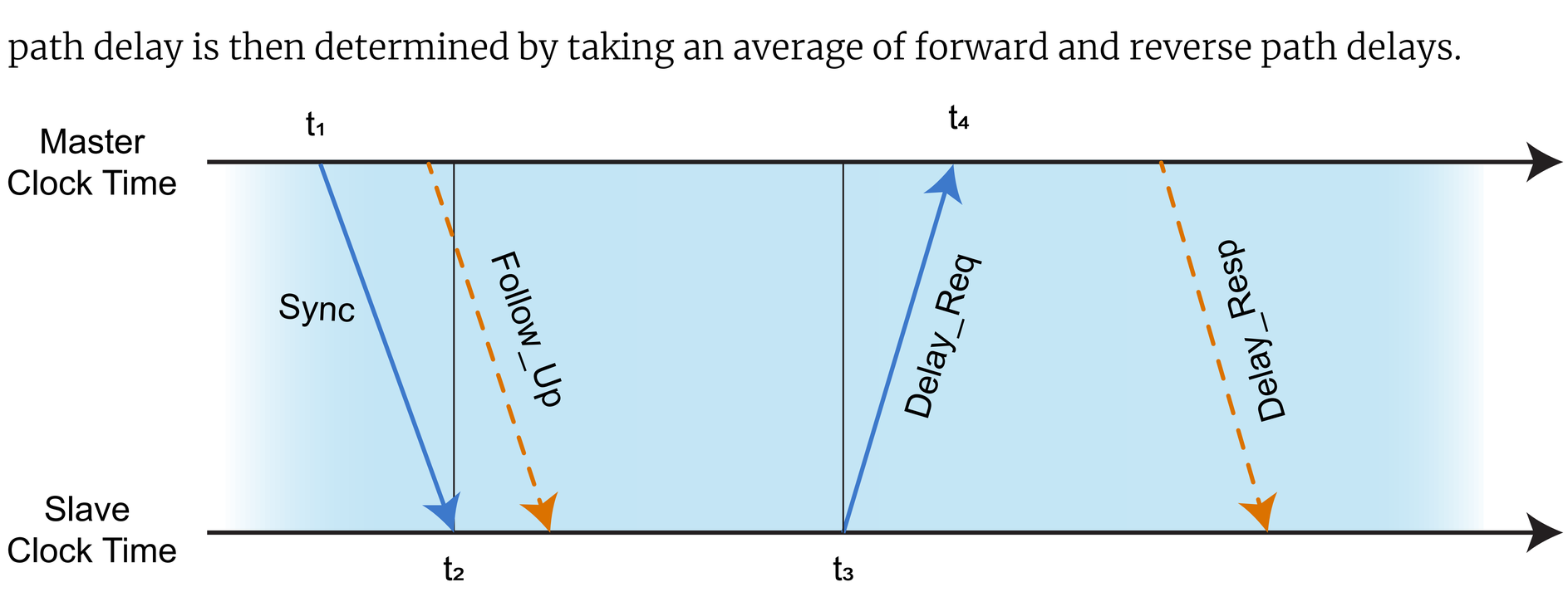
Event Messages	General Messages
Sync	Announce
Delay_Req	Follow_Up
Pdelay_Req	Delay_Resp
Pdelay_Resp	Pdelay_Resp_Follow_Up
	Management
	Signaling

PTP message types

$$Time_{diff} = Time_{Slave} - (Time_{Master} + PathDelay)$$

Since the path delay in forward and reverse paths (message transmission path) may not be identical, the path delay is calculated in the reverse direction (from Slave to Master) using Delay_Request and Delay Response messages.

The path delay is then determined by taking an average of forward and reverse path delays.



1. Master Clock sends a ‘Sync’ message.
   - In one-step mode, the Sync message carries the timestamp t1
   - In two-step mode, the subsequent follow-up message carries the timestamp t1
2. Slave receives the ‘Sync’ at timestamp t2.
3. Slave sends a ‘Delay Request’ message. It records the timestamp t3 when this message leaves the slave.
4. Master receives a ‘Delay Request’ message and records the timestamp t4 when the message was received.
5. Master responds with a ‘Delay Response’ message carrying the timestamp t4.

Slave calculates the Mean Path Delay.

Mean Path Delay = ((t2-t1) + (t4-t3))/2

Master Offset = t2-t1-MeanPathDealy = ((t2-t1)-(t3-t4))/2

As discussed earlier, time is simply an 80-bit counter maintaining the seconds and nanoseconds elapsed since 1 Jan 1970. Every PTP node has its own ToD counter to maintain time. The ToD counter is fed with a clock signal (a square pulse for example). The rate at which the counter gets updated depends upon the frequency of the clock signal. Hence, the time of a PTP node can be made to run faster (or slower) by increasing (or decreasing) the frequency of the ToD counter’s input clock signal.

Instead of doing a coarse correction of time, a slave node adjusts the frequency of the clock to achieve synchronization. A frequency synthesizer (or DPLL) is used to generate a frequency that can be digitally programmed/controlled. This controlled frequency is fed to the ToD counter.

Hardware Timestamping vs Software Timestamping

An inaccuracy in the timestamping of incoming and outgoing PTP messages will lead to inaccuracy in the calculated path delay and master offset. So, The accuracy of timestamping is the key to getting better synchronization.

For an outgoing (or incoming) message, the uncertainty in the amount of time spent in the transmit path (or receive path) is higher if the timestamping is done in higher layers of the OSI model because the CPU uses scheduling mechanisms to accomplish the various tasks running in an operating system.

A PTP message timestamped at a point very close to the exit point of the port for outgoing messages (lower layers of OSI model) and very close to the entry point for the incoming messages has lesser uncertainty and hence produces better synchronization performance.

Hardware timestamping is a mechanism in which the timestamping for an incoming or outgoing message is implemented in the HW usually next to the PHY layer (of the OSI model). This can be accomplished by implementing the Timestamping block in an FPGA

Software Time stamping is a mechanism in which timestamping for an incoming or outgoing message is implemented in the SW usually in the upper layers (of the OSI model).

Software time stamping allows PTP to synchronize systems to within a few tens of microseconds. With hardware time stamping, PTP can synchronize systems to within a few tens of nanoseconds. For high-precision time synchronization of systems, hardware time stamping is preferred.

Egress and Ingress Delays in PTP messages

- **Egress Delay:** In the transmit direction, when a message arrives at the Time Stamping Unit, it is parsed and a time stamp is inserted. Then the packet passes through the PHY and leaves the physical port. The time taken by the PTP packet to traverse through hardware after getting timestamped is referred to as Egress Delay.

- **Ingress Delay:** In the receive direction, when a message arrives at the Time Stamping Unit, it is parsed and the time stamp is recorded. So, the packet passes through the PHY and some blocks in TSU before the timestamp is recorded. Ingress Delay is the time elapsed from the instance the PTP message entered the physical port till the corresponding timestamp is recorded.

Both the egress and ingress delays must be known and shall be compensated to accurately measure path delay. The ingress delay is subtracted from the received timestamp before feeding to the servo.

While the egress delay is added to the transmit timestamp before inserting the timestamp into the packet. By doing so, the PTP engine can accurately measure the path delay.

PTP Messages

Master and slave network devices are kept synchronized by the transmission of timestamps sent within the PTP messages. There are two types of messages in the PTP protocol: Event Messages and General Messages. Event messages are timed messages whereby an accurate timestamp is generated both at transmission and receipt of the message. General messages do not require timestamps but may contain timestamps for their associated event message.

PTP Profiles

ITU-T has defined certain PTP profiles to cater to different use cases

- ***ITU-T G.8265.1*** – Precision time protocol telecom profile for frequency synchronization
- ***ITU-T G.8275.1*** - Precision time protocol telecom profile for phase/time synchronization with full timing support from the network
- ***ITU-T G.8275.2*** – Precision time protocol telecom profile for phase/time synchronization with partial timing support from the network.

A profile also defines the allowed clock types, transmission modes (Ethernet or UDP), Unicast or Multicast addressing, Message rates, etc.,
Full timing support indicates that all the nodes in the network are PTP aware.
Partial timing support indicates that NOT all nodes in the network are PTP aware.

Key Components of PTP

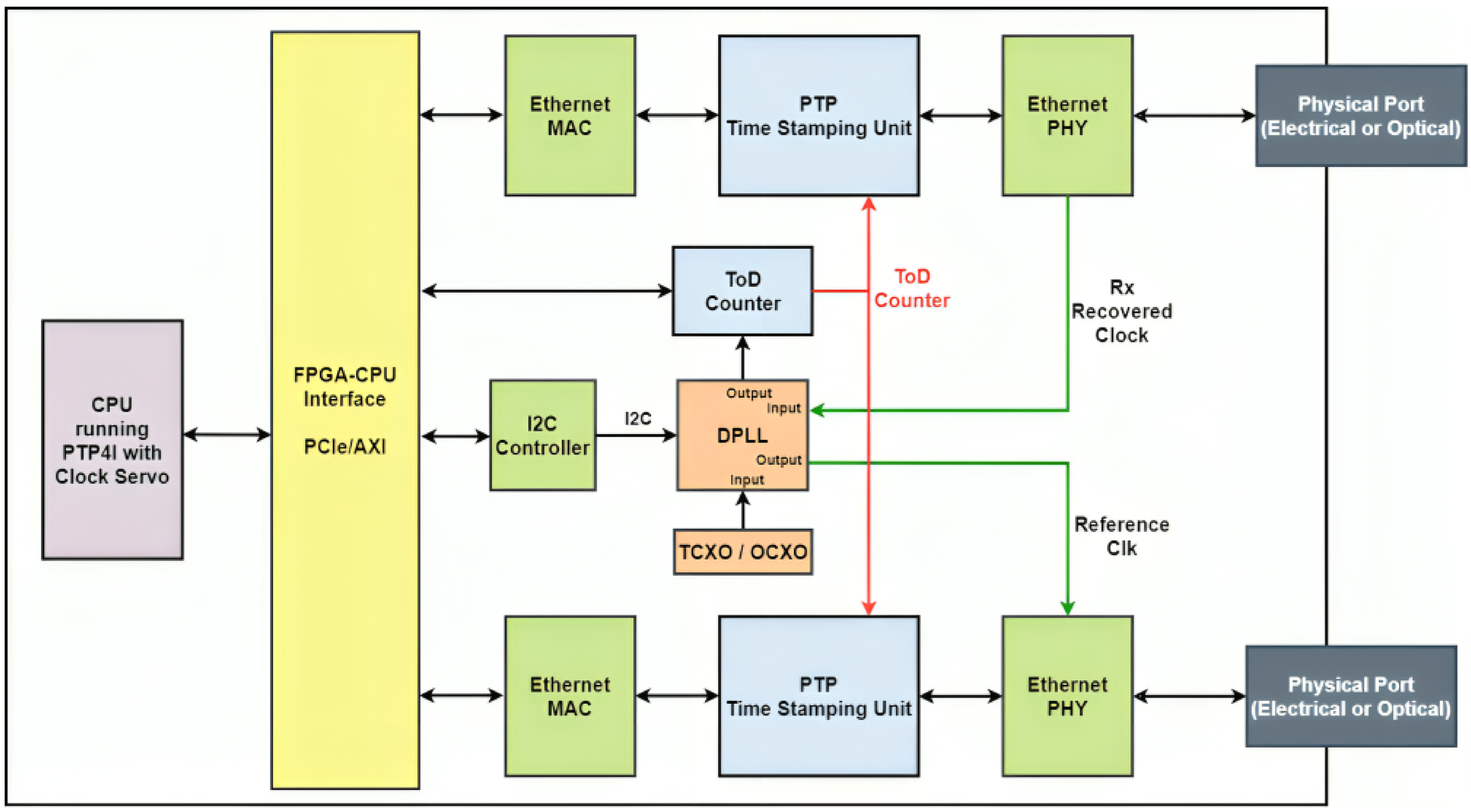
PTP Engine - Responsible for generating the PTP messages for transmission, processing the incoming PTP messages, and calculating the master offset and frequency offset, etc.,

LinuxPTP is an Opensource implementation of the PTP protocol, a PTPv2 implementation according to the IEEE standard 1588 for Linux. The LinuxPTP package includes the ptp4l and phc2sys programs for clock synchronization.

The ptp4l is the PTP engine and phs2sys is the engine responsible for synchronizing the PTP hardware clock with the Linux system clock.

- **Clock Servo -** Responsible for tuning the clock frequency to achieve synchronization. ptp4l provides an inbuilt clock servo.
- **TimeStampingUnit -** Responsible for timestamping the incoming and outgoing messages, recalculating the UDP checksum, FCS as required after inserting timestamps.

FPGA + SW Based Implementation



Going from left to right

1. **CPU**
   - Desktop CPU (with Linux-based OS) that can run LinuxPTP
   - Any Embedded CPU (like ARM cortex) running embedded Linux
   - The CPU can also be a softcore processor inside an FPGA.

2. **CPU - FPGA Interface** - the PTP message received by the FPGA is transferred to the CPU over this interface. There could be other configuration and control messages that will be exchanged over this interface.

3. **Ethernet MAC**

4. **PTP Timestamping Unit**

- Packet Parser to identify the PTP packets
- Timestamp trigger generator
- Timestamp inserter

5. **Ethernet PHY**

6. **ToD counter**

- It has the clock signal input from DPLL
- Provides the 80-bit ToD output upon receiving a trigger
- The output is an 80-bit counter value indicating the time at the instance of receiving the trigger

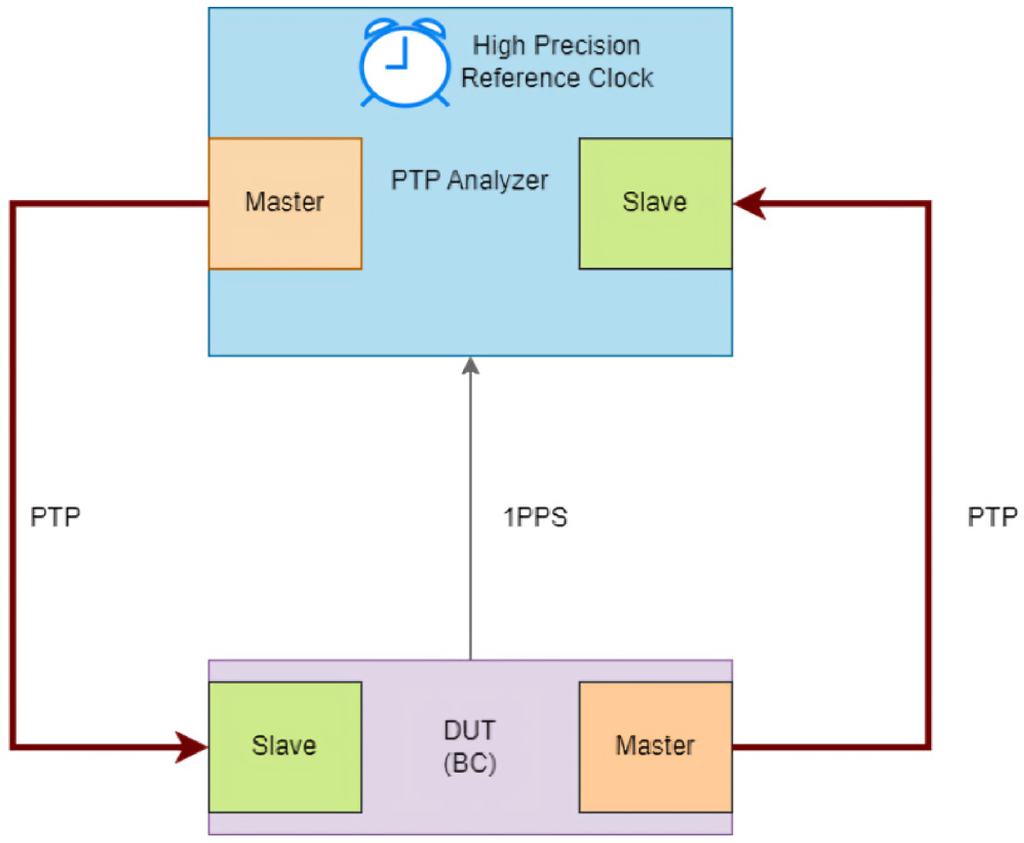
7. **DPLL:**

- Digital Phase Locked Loop.
- This can also referred to as a clock synthesizer.
- It can generate any frequency output through a digitally controlled interface like I2C or SPI.
- When tuning the clock, this module is responsible for providing the input clock signal to the ToD counter

8. **TCXO / OCXO :**

- High precision and high stability standalone clock input for the DPLL.
- In case of Loss of PTP, this input helps the clock maintain the time with minimal drift. This state of a PTP clock is referred to as Holdover.

Test setup for a Boundary clock



☞ A typical setup for testing a Boundary clock is shown above

☞ A PTP analyzer has two ports

- ***Master Port:*** Provides time over PTP to the slave port of the BC
- ***Slave Port:*** This is not a true slave port. It only derives time from the Master port of BC but does not tune the reference clock.

☞ The Analyzer provides PTP data to the slave port of BC

☞ BC tunes its clock reference

☞ BC provides PTP data to the slave port of the Analyzer

☞ The analyzer extracts the time, compares it with the reference clock, and provides the required performance metrics

Performance tests (as per ITU-T G.8273.2)

1. **Noise generation:**

The noise generation of a T-BC and a T-TSC represents the amount of noise produced at the output of the T-BC/T-TSC when there is an ideal input reference packet timing signal.

2. Noise Tolerance:

The noise tolerance of a T-BC/T-TSC indicates the minimum dynamic time error level at the input of the clock that should be accommodated while:

a. not causing any alarms;
b. not causing the clock to switch reference;
c. not causing the clock to go into holdover.

3. Noise Transfer

Measures how Time error on the input is transferred to the output

The transfer characteristic of the T-BC/T-TSC determines its properties regarding the transfer of time error from the PTP input interface to the PTP and 1PPS output interfaces.

4. Noise Transient response

PTP output and 1PPS output transient response due to rearrangement of physical layer frequency transport and PTP network

5. Holdover

Measure the maximum excursions in the PTP and 1PPS output signal during the loss of PTP input and/or physical layer frequency input.

Conclusion

PTP has an increasing demand in various applications. The IEEE standards enabled the protocol future ready, scalable, and cost-effective. The working principle is simple yet very elegant and can fulfill the stringent synchronization needs. The implementation using software to run the PTP stack and a Time stamping unit in FPGA provides a very flexible solution. With some insights into how PTP works and the key factors affecting the performance, the designer can achieve high-quality synchronization. The testing tools are also readily available to evaluate the performance and thereby address issues to improve upon the design.

References

- *IEEE Std 1588™-2008*
- *IEEE Std 1588™-2019*
- *IEEE 802.1AS*
- *ITU-T G.8265.1*
- *ITU-T G.8275.1*
- *ITU-T G.8275.2*
- *ITU-T G.8273.2*
- *Test Guide CX3009 v3.0 from Calnex*

logic fruit Technologies

Thank You!

Does anyone have any questions?

Gurugram (Headquarter)

806, 8th Floor
BPTP Park Centra Sector–30,
NH–8 Gurgaon – 122001
Haryana (India)

info@logic-fruit.com

+91-0124 4643950

Bengaluru (R&D House)

Sy. No 118, 3rd Floor,
Gayathri Lakefront,
Outer Ring Road, Hebbal,
Bangalore - 560 024

sales@logic-fruit.com

+91 80-69019700/01

United States (Sales Office)

Logic Fruit Technologies
INC 691 S Milpitas Blvd
Ste 217 (Room 9)Milpitas
CA 95035

info@logic-fruit.com

+1-408 338 9743