Technical White Paper

Time Sensitive Networking With Ethernet PHY

Melissa Chang and Geet Modi

ABSTRACT

Time synchronization plays a crucial role in modern vehicles, enabling many automotive applications. These applications include synchronizing the activation of lights like front and rear indicator, making sure audio playback from multiple speakers is correctly timed, using coordinated sensor inputs like accelerometer and microphone output to cancel road noise, synchronizing the GNSS receiver to the v2x transceiver, and simultaneously operating LiDAR and radar sensors for precise area scanning.

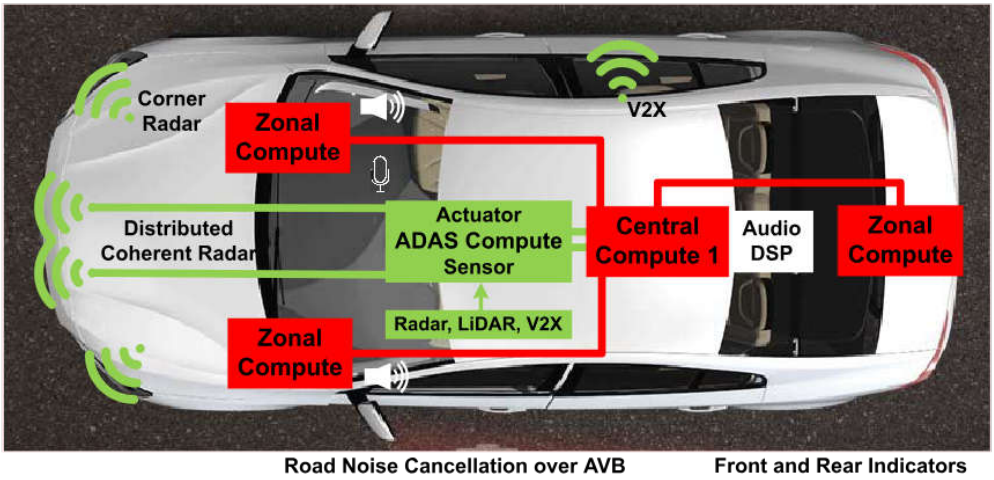

Figure 1-1. Automotive Architecture, focusing on ECUs where Time Synchronization is required

One method of introducing time synchronization to the aforementioned applications is to implement the open standard IEEE 802.1AS with Ethernet. IEEE 802.1AS comes from Time Sensitive Networking (TSN), or a set of standards that define a method for designing systems with “packet transport with bounded latency, low packet delay variation, and low packet loss” [1] . IEEE 802.1AS is one of the TSN standards that specifies the protocol and procedures used to make sure that the synchronization requirements are met for time sensitive applications with a millisecond to nanosecond synchronization accuracy. This whitepaper gives a brief overview of the applications that require time synchronization, how IEEE 802.1AS works, and how TI Ethernet PHYs DP83TG721, DP83TC817, and DP83TC818 implement IEEE 802.1AS for precise synchronization accuracy and reduced software overhead.

Table of Contents

List of Figures

Trademarks

All trademarks are the property of their respective owners.

1 The Role of Time Sensitive Networking in Automotive Applications

Modern vehicles use a variety of sensors that require precise synchronization, the level of synchronization required is application specific

One simple example is opening the trunk automatically using motors. To verify that the trunk opens smoothly, the left and right motors need to operate simultaneously. Since motors can be controlled by Pulse Width Modulation (PWM) signals, the motors can operate simultaneously with synchronous PWM inputs. The role of IEEE 802.1AS is to help generate the synchronous PWM waveforms. The synchronization accuracy requirement for this application can be in milliseconds.

Another good application for time synchronization is car audio systems. For high quality spatial audio experience, the front and rear multiple speaker's audio playback needs to be fully synchronized. This can be achieved by synchronizing the audio play back ECUs and Central Signal Processing/mixer responsible for generating the audio streams for playback. IEEE 802.1AS can be used to achieve the time synchronization between front, rear amplifiers and Central Signal Processors and compensate for the varying path delays. Signal processors transmit audio packets while embedding the presentation time, the time at which audio packets need to be played out on audio amplifiers. The audio amplifier, despite different path delays, plays out the audio packets at the same synchronized time. The typical synchronization accuracy required for this application is in microseconds.

A very good application of precise time synchronization is ADAS sensor synchronization. These sensors (radar, LiDAR, camera) are physically distributed across the car to sense the varying environment variables (obstruction, speed of nearby object, and so forth). The scanned information is provided to ADAS compute to make holistic understanding of 360° view of the car. The tighter the data from these sensor is synchronized, the better the ADAS compute is able to detect and handle real obstructions. To operate the radar synchronously, each radar sensor shall initiate the scan of the environment (chirp) at the same time (nanosecond accuracy). Timing mismatch can lead to blurring in the final image. Each radar sensor can receive a synchronous pulse per second (PPS) reference input to chirp at the same time. The role of IEEE 802.1AS is to help generate the synchronous PPS signal.

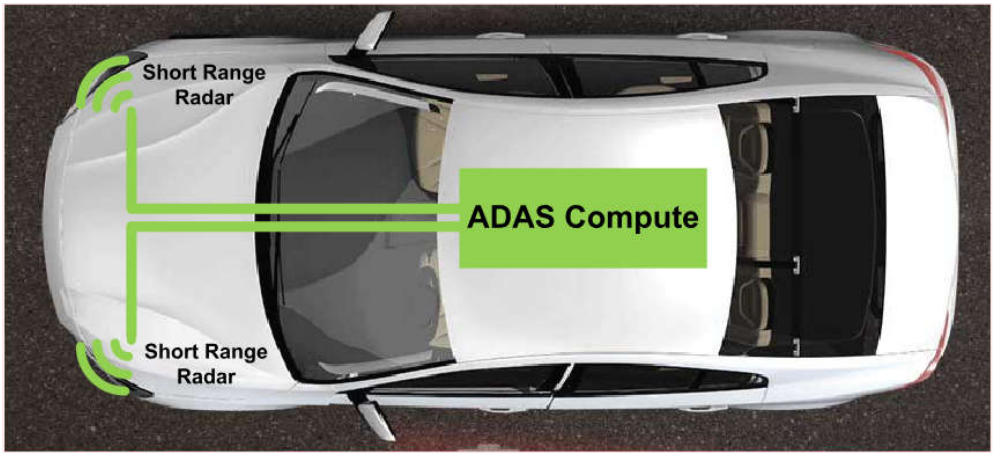

Figure 1-1. Extended Aperture Radar Example

Whether the application requires generating waveforms or transmitting timestamps over a time sensitive network- all nodes must operate based on a synchronized reference clock. IEEE 802.1AS defines the procedure and protocol to implement the synchronized reference clock. However, the accuracy of time synchronization is dependent on the hardware architecture implementing the 802.1AS. In the following sections, an overview of IEEE 802.1AS procedures is provided along with some of the architecture choices used by TI Ethernet PHY to achieve precise time synchronization.

2 Generalized Precision Time Protocol Algorithm Overview

IEEE 802.1AS implementation in ECUs typically requires two components:

- Timestamping of IEEE 802.1AS packets (also known as PTP packets)
- Using timestamps generated by the first component for computing the path delay, countering the offset, and ppm drift compensation.

The first component is typically implemented in hardware for better synchronization accuracy while second component is typically implemented in software also known as gPTP.

The following section outlines how the gPTP algorithm achieves time synchronization with the reference clock of two ECUs, one leader and follower. In IEEE 802.1AS, this reference synchronized clock is also known as the Time of Day reference clock or Wall Clock. The Wall Clock (see Figure 2-1) can be modeled as a counter that toggles at a fixed increment.

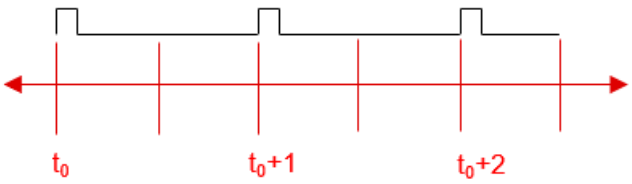

Figure 2-1. Wall Clock Model

There are three main tasks gPTP as to perform. The first task is to synchronize the Wall Clock of the follower ECU to the leader ECU, since each ECU can start at a different time. Figure 2-2 shows, for two wall clocks to be synced, the delay, Δt, must be the same.

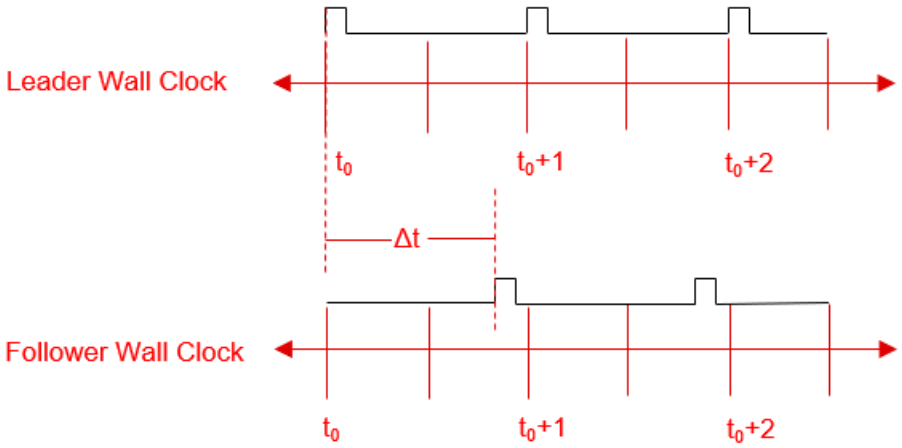

Figure 2-2. Delay Offset Between Two Clocks

The second task of gPTP is to correct for the PPM drift of the Wall Clock source. Every clock has some PPM clock drift that causes the frequency to shift slightly so even if the delay is aligned, the clock ppm drift can introduce another delay, Δf, between cycles.

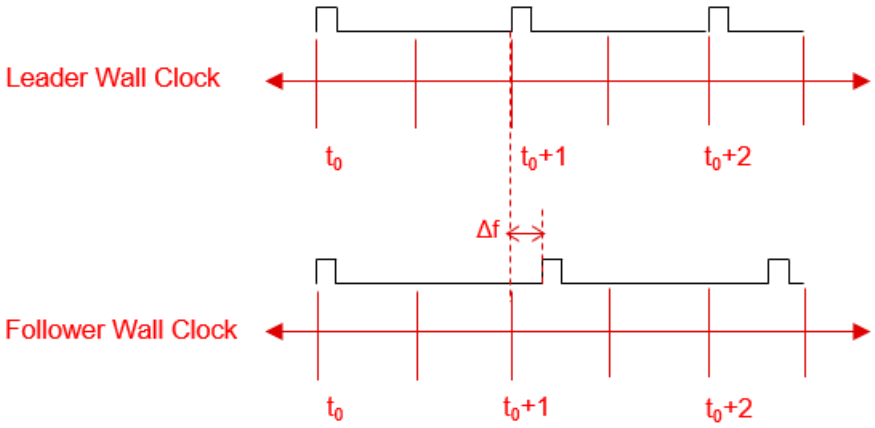

Figure 2-3. Frequency Offset Between Two Clocks

To solve for this difference in counters (time offset) and frequency (clock drift) between two clock signals, the leader ECU and follower ECU can exchange a set of timestamps to calculate the difference. When calculating the delay between the two Wall Clocks, there is a path delay caused by the time the timestamps take to travel between leader and follower.

The third task of gPTP is to calculate and account for this path delay.

Figure 2-4 shows the set of timestamps exchanged between the leader and follower to calculate path delay, time offset, and clock drift compensation.

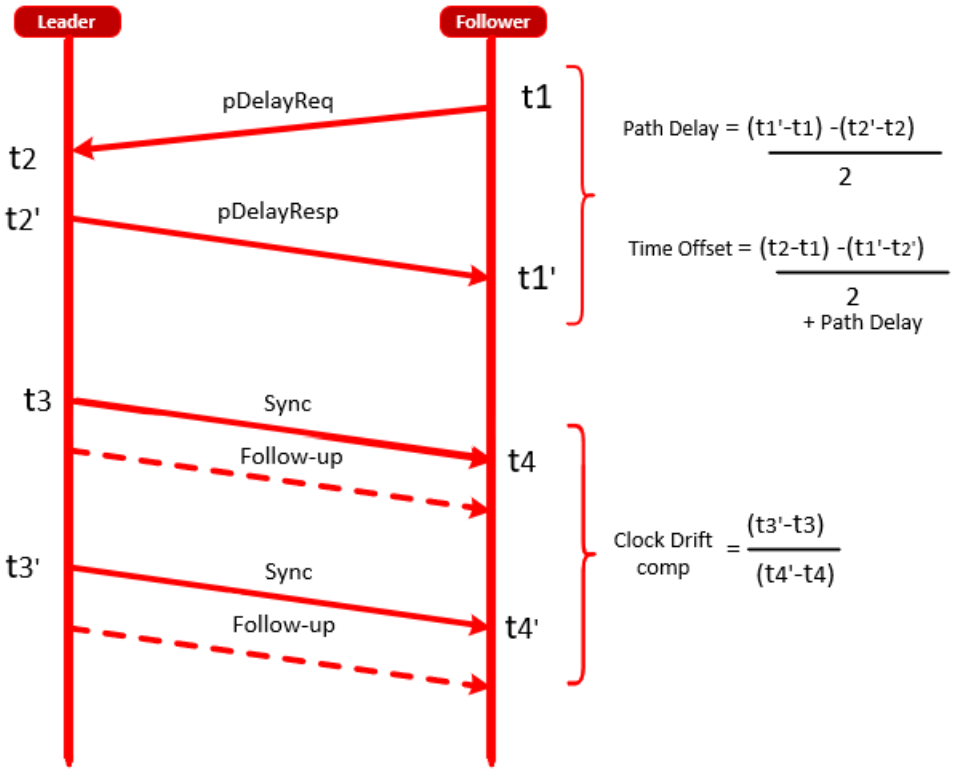

Figure 2-4. gPTP Time Offset and Clock Drift Equations in Two-Step Synchronization

2.1 gPTP Timetamping Handshake Process

To sync with the PTP leader, the PTP follower must adjust the offset to match the leader. This process is initiated by the follower sending out a peer delay request, pdelay_req, which is timestamped at t_1. Then, the leader receives pdelay_req at t_2 and responds with a peer delay response, pdelay_resp, at $t_{2'}$, which the follower receives at $t_{1'}$. During the handshake, $t_{2'}$ and t_2 are forwarded to the follower. Once timestamps t_1, t_2, $t_{2'}$, and $t_{1'}$ are recorded by the follower, the timestamps can be used to calculate to the offset and path delay to adjust the follower's wall clock to match the leader's wall clock using the equation shown in Figure 2-4. Note that the equation for path delay assumes a symmetrical path of travel.

The follower can also adjust the frequency to match the leader. The leader continuously sends out sync messages at t_3, which is received by the follower at t_4. The succeeding timestamps can be represented as $t_{3'}$ and $t_{4'}$ respectively, as shown in Figure 2-4. The frequency drift can be calculated using the equation shown from Figure 2-4. If the ratio is 1, no adjustment is made. If the ratio is 0.9995, each succeeding cycle can be 0.9995 multiplied by the period of the previous cycle.

A follow-up message is also sent to include the time the original sync message was actually transmitted because there is some delay between when the leader initiates the sync messages to when the message is actually sent.

The leader and follower clocks can stay synchronized if each timestamped packet always take the same time to travel across the network. However, in reality, packet travel times vary due to unpredictable delays from the processor and network. This makes synchronization more challenging, but there are ways to address these challenges mentioned in the next few sections.

Note

For applications with more than two processors boundary clocks can be used to distribute the PTP protocol across the network. For more details on how boundary clocks work, see *IEEE 1588 Boundary Clock and Transparent Clock Implementation Using the DP83640*.

3 Methods of Implementing gPTP: Timestamping Location

For gPTP to work, the systems that are synced with each other must each have a common wall clock source and a way to timestamp egress and ingress packets. The point in the system where these packets are timestamped directly impacts the synchronization accuracy.

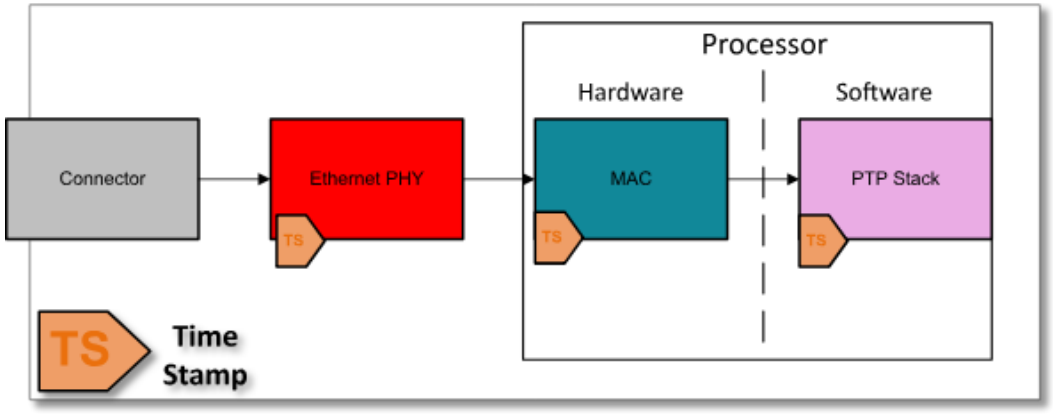

Figure 3-1. Different Kinds of Timestamping in a System

The closer the packet is timestamped to the cable, the more indeterministic latencies are accounted for and the more accurate the synchronization is.

For applications that only require millisecond level synchronization such as lighting and trunks, packets can be timestamped in the software stack. This method is simple but introduces invariability due to software processing delays.

Applications requiring microsecond level precision, such as audio and road noise cancellation, can use hardware level timestamping on the MAC level. This approach reduces latency variability compared to software timestamping, providing improved accuracy.

The most accurate way to timestamp packets is directly at the Ethernet PHY level, which can have yield nanosecond to sub nanosecond sync accuracy for ADAS sensor applications. Texas Instrument's DP83TG721 and DP83TC817/8 PHYs with IEEE 802.1AS support offer PHY level timestamping.

4 Fixed Latency and Recovered Clock Modes

In addition to supporting IEEE 802.1AS HW timestamping, The DP83TG721 and DP83TC817/8 have additional features that account for indeterministic latencies that occur as a result of the PHY.

For ADAS sensor applications, both devices offer the option of using the recovered MDI clock as the Wall Clock source. Using the recovered clock, the system can eliminate the ppm drift and associated compensation enabling better synchronization performance. Recovered clock also does not require continuous adjustments, freeing up software traffic and overhead. As a result, maximum throughput can be used with minimal affect on sync accuracy.

Figure 4-1 shows examples of point-to-point PPS synchronization using the local clock and using the recovered clock source on the DP83TG721. From the scope shots, using the recovered clock results in a 4.5ns delay and 0ns of jitter while using the local clock results in 3ns delay and 27ns of jitter.

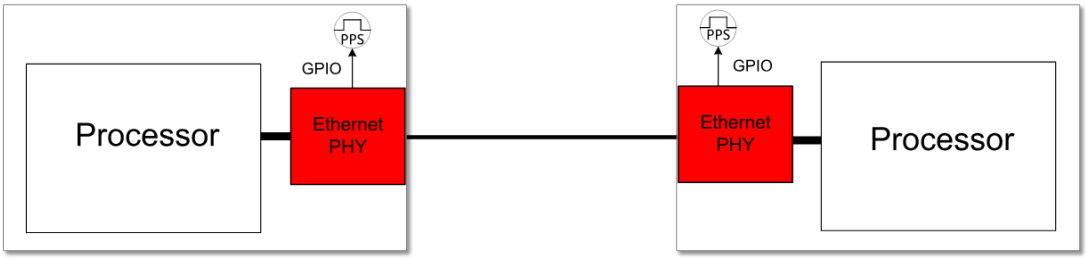

Figure 4-1. DP83TG721 PPS Synchronization Setup: GPIO Pins are Generating PPS Signals From the PHY While PTP is Running in the Background

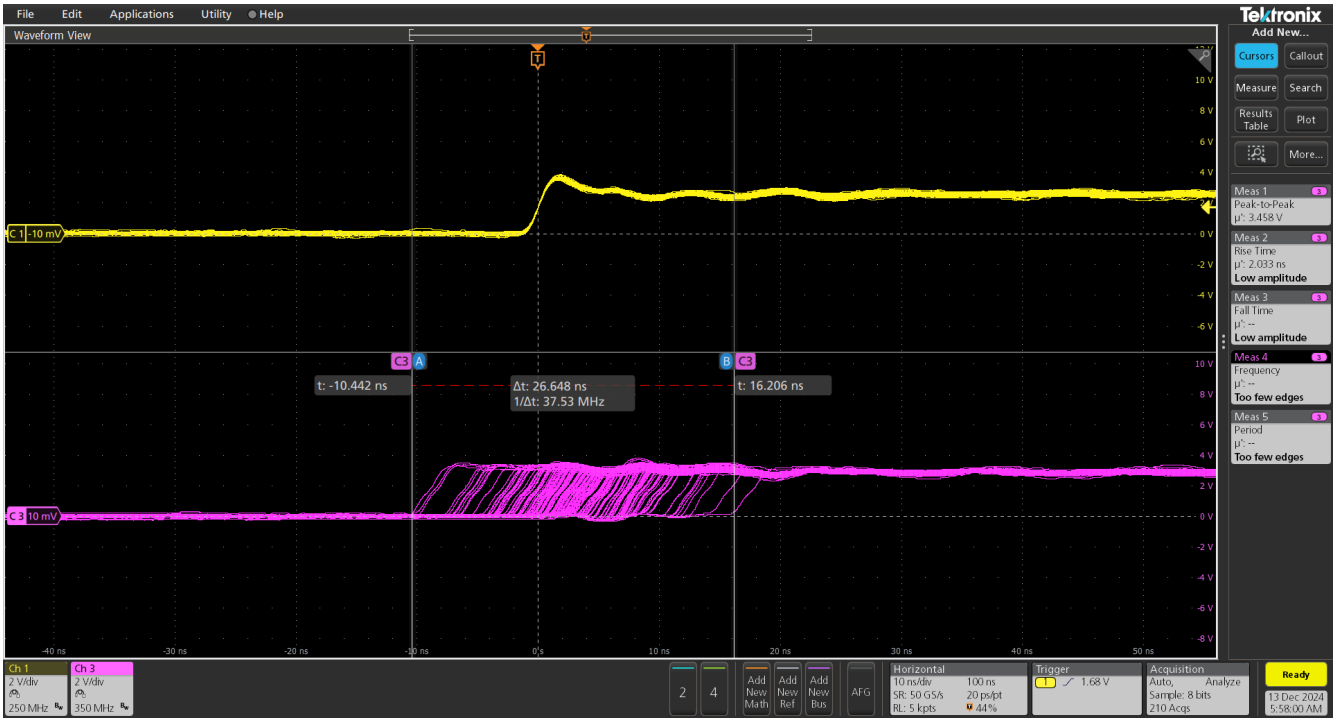

Figure 4-2. DP83TG721 PPS Synchronization With Local 125MHz Clock as PTP Clock Source

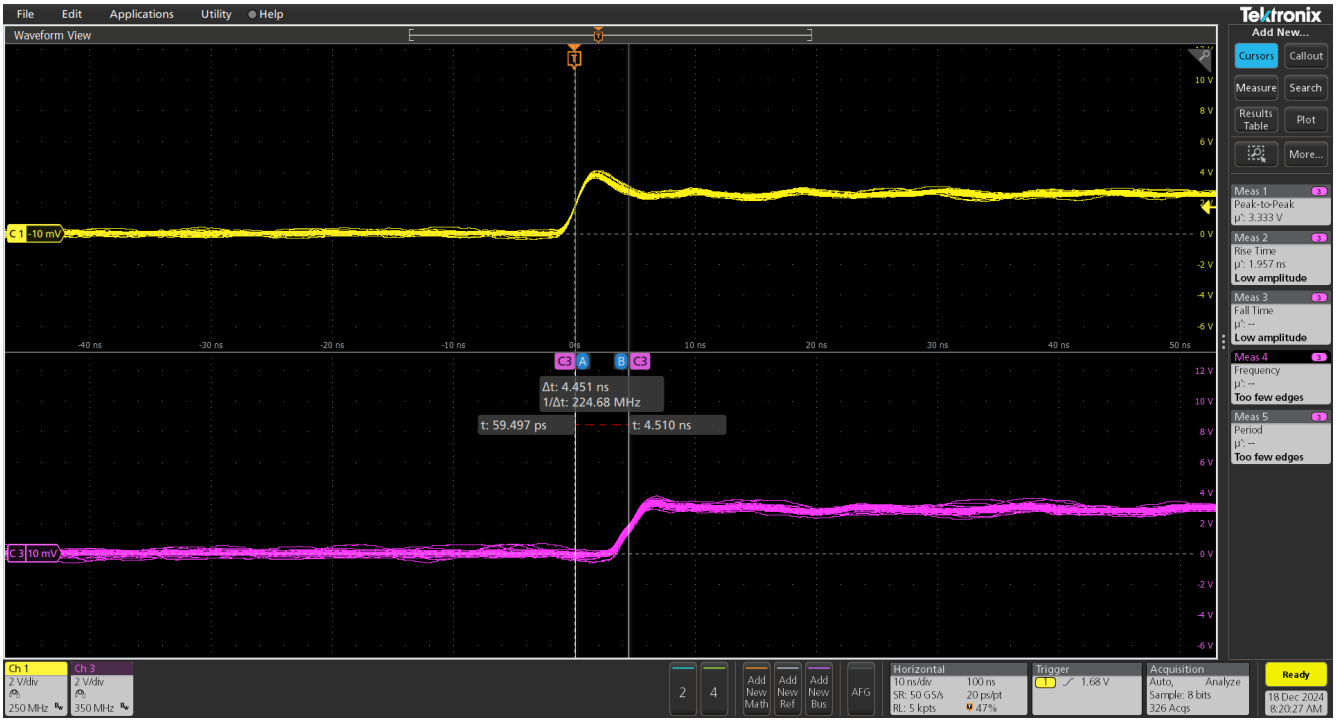

Figure 4-3. DP83TG721 PPS Synchronization With MDI Recovered Clock as Clock Source

For TI's 100Base-T1 IEEE 802.1AS, a fixed latency feature has also been added to account for the latency variations that occur in the PCS layer of the PHY. The lower speed PHY (100Base-T1) PHY have larger latency variation due to slower clock speeds. Detailed architecture level considerations are needed to offer precise synchronization. These architecture level considerations have been added to minimize the effect of latency variation.

Below are some example measurements that had been taken with and without fixed latency mode with the DP83TC817. The scope shot using fixed latency mode shows 21ns of jitter while the scope shot without fixed latency shows 50ns of jitter, demonstrating a much better sync accuracy in fixed latency mode. By combining fixed latency mode and recovered clock mode, an even better synchronization accuracy is achieved with of 0ns of jitter and only 6.3ns of delay.

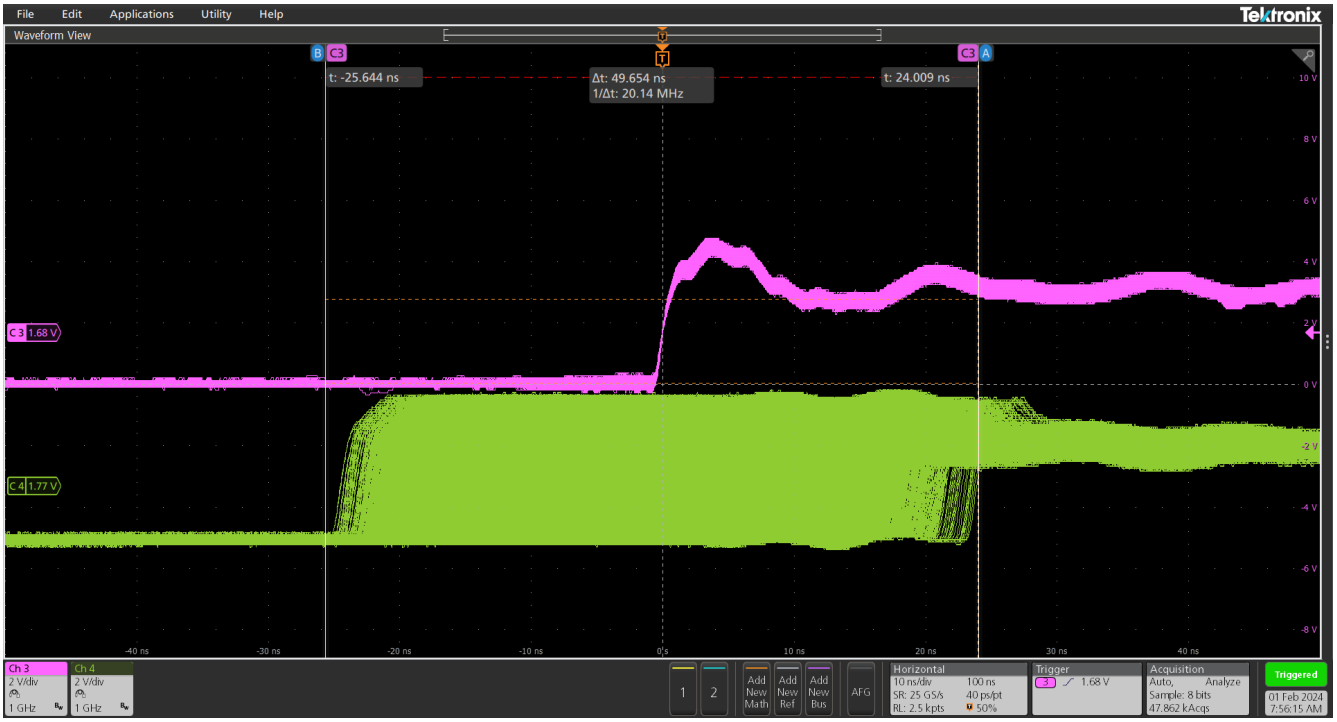

Figure 4-4. DP83TC818 PPS Synchronization With Local 250MHz Clock and Fixed Latency Disabled

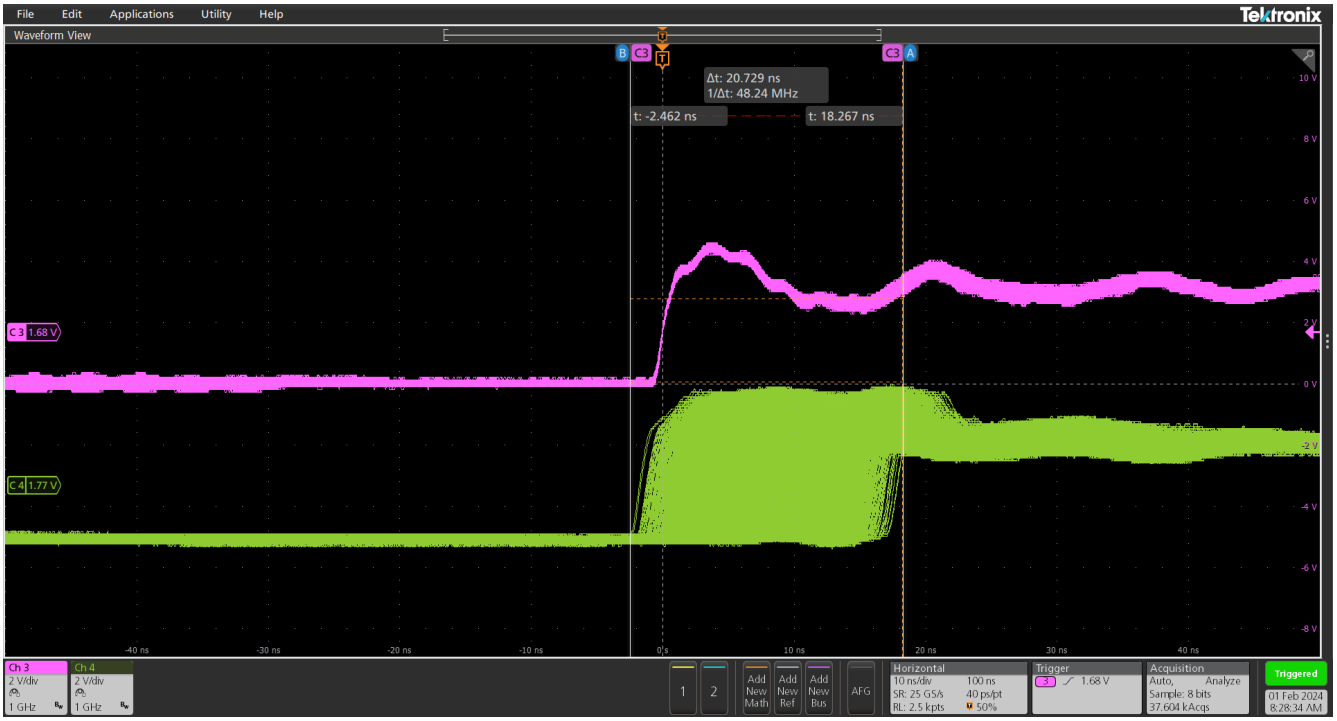

Figure 4-5. DP83TC818 PPS Synchronization With Local 250MHz Clock and Fixed Latency Enabled

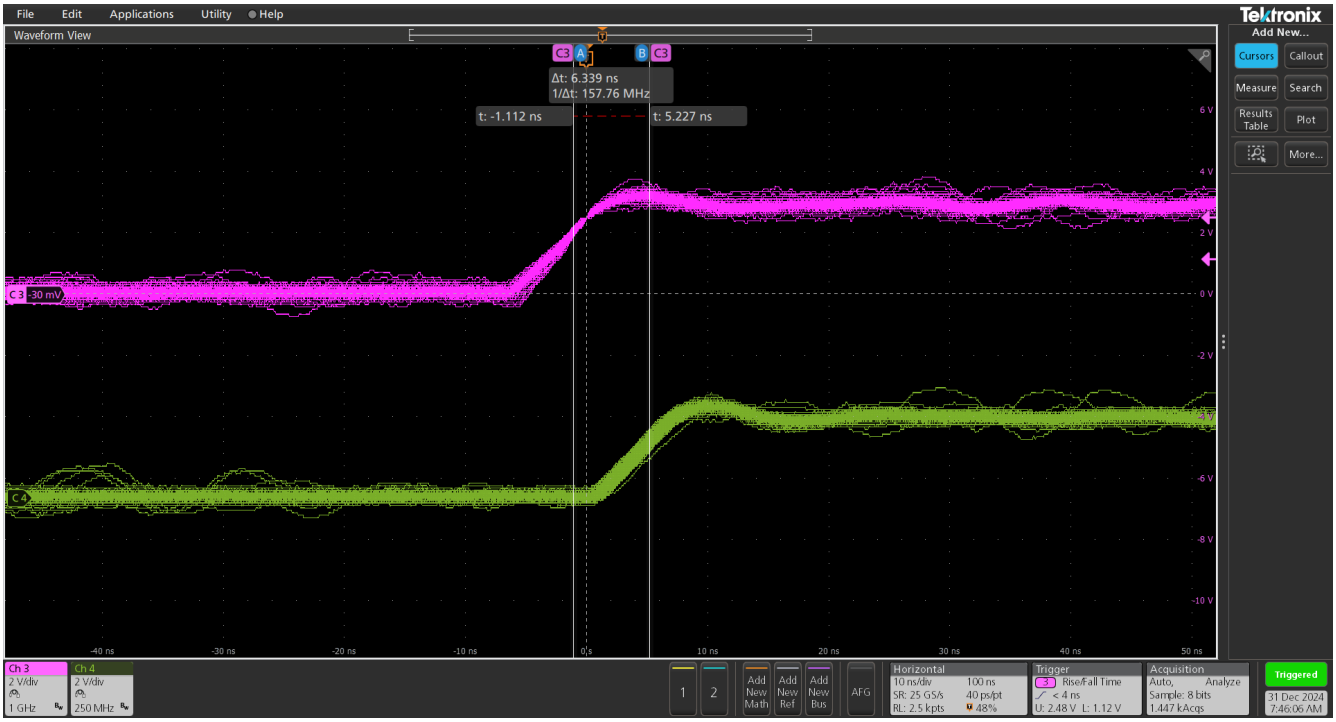

Figure 4-6. DP83TC818 PPS Synchronization With Recovered 200MHz Clock and Fixed Latency Enabled

5 Event Triggers and Monitors

DP83TG721 and DP83TC817/8 also support Event Triggers and Monitors on the PHY's GPIO pins to offload some of the processor overhead of generating these for TSN applications. Event Triggers are GPIO generated PWM waveforms that are aligned with the Wall Clock. These can be used to generate synchronized pulse per second (PPS) waveforms, or 1Hz PWM waveforms.

Event monitors are input waveforms to GPIO pins that can be timestamped for every rising or falling edge. These features are useful for applications such as below, where you want to synchronize an input coming from some receiver to an output going to a transceiver that is connected via Ethernet.

Figure 5-1 shows PPS input and PPS output are synchronized through PTP. Both are received and generated by the processor.

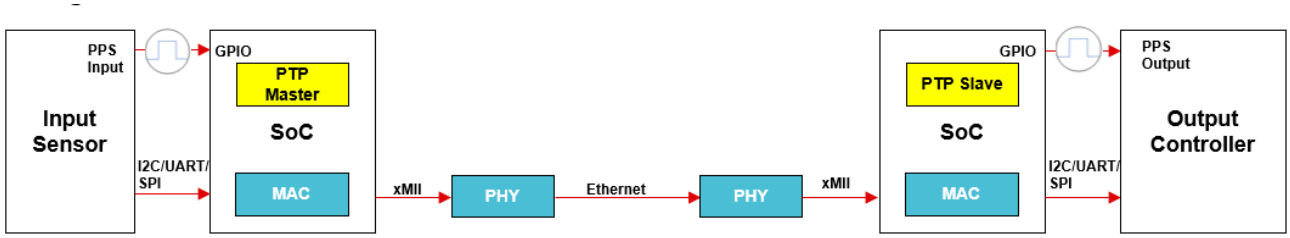

Figure 5-1. Example PPS Application

Figure 5-2 shows PPS input and PPS output are synchronized through PTP. Both are received and generated by the PHY, offloading the processor overhead from the previous figure.

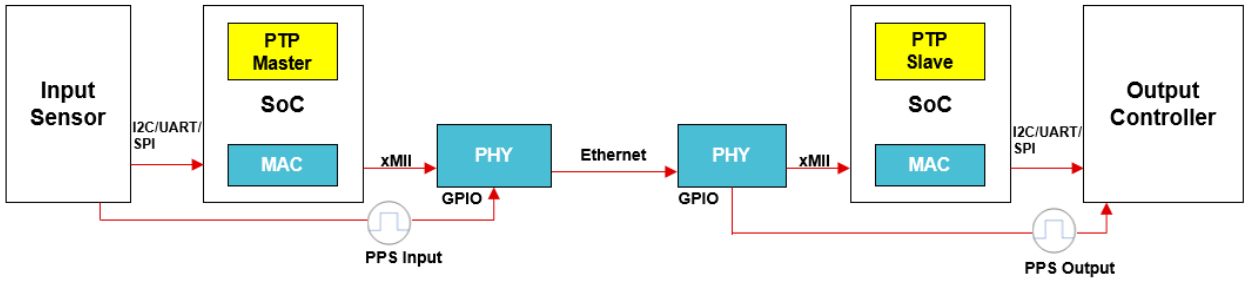

Figure 5-2. Application of PHY Event Monitor and Triggers

6 Simplified gPTP Integration

When IEEE 802.1AS is implemented with PHY level timestamping, the PTP timestamped messages and associated PTP sequence IDs are stored within the PHY. Accessing these packets using the traditional MDC/ MDIO serial interface can take microseconds to a few milliseconds and is sometimes complex to integrate with 802.1AS software stack running in the SoC.

To ease the integration with gPTP, TI PHYs offers additional two ways to provide PTP timestamps to host SOC.

- Appending the timestamp in the PTP message before being forwarded to the ECU through the RX interface.
- A packet-based status mechanism that allows the PHY to generate in-band Ethernet frames with timestamps to the SoC. The packet, called a **PHY Status Frame**, can be used provide transmit PTP packet timestamps, receive PTP packet timestamps, event timestamps, and trigger conditions.

These additional capabilities allow SoC to access the PTP messages directly from the IEEE 802.1AS software stack as opposed to having to access the messages through MDIO/MDC, making the integration of gPTP easy and efficient.

7 Conclusion

Time synchronization is essential for numerous automotive applications. IEEE 802.1AS serves as a reliable method for achieving time synchronization. There are multiple ways to implement IEEE 802.1AS, but timestamping at the PHY level delivers the most precise level of synchronization. TI's DP83TG721 and DP83TC818/7 offer additional features such as fixed latency mode and MDI recovered clock for improved synchronization, event trigger and monitor features reduce the processor overhead for these TSN applications, and the generation of PHY status frames for simplified gPTP integration.

8 References

1. IEEE Std 802.1AS™-2020 "IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications"
2. Texas Instruments: *IEEE 1588 Boundary Clock and Transparent Clock Implementation Using the DP83640*
3. Texas Instruments: *DP83TG721x-Q1 1000BASE-T1 Automotive Ethernet PHY with Advanced TSN and AVB Data Sheet*
4. Texas Instruments: *DP83TC818S-Q1 Precise and Secure 100BASE-T1 Automotive Ethernet with AVB Clock Generation, IEEE802.1AE MACsec, IEEE802.1AS and TC10 Sleep-Wake Data Sheet*
5. Texas Instruments: *DP83TC817S-Q1 Precise and Secure 100BASE-T1 Automotive Ethernet with IEEE802.1AE MACsec, IEEE802.1AS and TC10 Sleep-Wake Data Sheet*
6. Texas Instruments: *How to Implement IEEE 1588 Time Stamping in an Ethernet Transceiver*

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265