

AM65xx Time Synchronization Architecture

Jian Wang

ABSTRACT

This application report describes the multi-protocol time synchronization architecture of the AM65xx-class of devices. It starts with an overview of time-sync protocols and their applications in automation and industrial control systems. Then, introduces hardware components of the time-sync network in the device. Followed by, two example embedded systems where multi-protocol time-sync are implemented based on this architecture.

Contents

1	Introduction	2
2	AM65xx Time Sync Architecture	4
3	Time-Synchronization Examples.....	8
4	Summary	12
5	References	12

List of Figures

1	Example of a Control Module With Multiple Time Bases	3
2	Multi-Protocol Time Synchronization Network in AM65xx	4
3	Time-Sync Router (TSR) and Subsystem Modules	5
4	Functional View of CPTS Subsystem	6
5	PCIe PTM Functional View.....	7
6	IEP Timers and Clock Selection	7
7	GTC and Clock Selection.....	8
8	Time-Sync Use Case Example 1: AM65xx as Time Masters.....	9
9	Time-Sync Use Case Example 2: Time Synchronization Across Two Devices	10

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

In control and automation systems, applications and communications are often required to be synchronized across the network or across devices within the subsystem. Each level of synchronization has its own precision requirements. For example, a cloud-based factory may require its "Wall Clock" to be synchronized via GPS, across its global locations; or a control network may require all devices in the network to be synchronized to a "Working Clock", for time-triggered frame transmissions; or, a subsystem with multiple devices and clock sources will be synchronized to a common time-base, sometimes referred to as "System Time".

[Table 1](#) shows commonly used time synchronization protocols defined on different network interfaces. These protocols govern inter-operability of network components when they are implemented. However, when a device supporting multiple types of interfaces is used, the system may require the device to synchronize time-base across these interfaces, hardware architecture support may be needed to facilitate software implementation.

Table 1. Commonly Used Time-Synchronization Protocols

	Description	Released	Best Achievable Accuracy	Applications	TI Devices
GPS	GPS Satellites	---	100 μ s-1 ms	Wire- and wireless networks	---
IETF NTP	Network Time Protocol	1985	Tens of Milliseconds<1ms LAN	Internet	Software
IEEE 1588 2002	Precision Time Protocol	2002	<1 μ s	LAN	K2H, K2L, AM3x/4x/5x, AM65x
IEEE 1588v2	Precision Time Protocol V2	2008	<1 ns	LAN	K2H, K2L, AM3x/4x/5x, AM65x
IEEE 802.1AS	Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks	2011	<1 ns	LAN	K2H, K2L, AM3x/4x/5x, AM65x
PCIE PTM	Precision Time Measurement	2013	1 ns	Real-time system clock synchronization	AM65x
SyncE	Synchronous Ethernet	~2010	0.01 PPM	Wireless etc	K2H, K2L
SyncR (TI internal naming)	Synchronous Radio I/F (CPRI)	---	---	Wireless	K2L

Figure 1 shows a simplified control module, where three controller devices are deployed. Device A is the host where Devices B and C are connected to it via PCIe ports. Device B and C are network interface cards that is connected to A. Additionally, Device B connects to a GPS receiver that sends out a pulse train for the Global Time (Wall Clock). Device C receives a Working Clock via the IEEE 802.1AS protocol. In later sections of this document, it is described how these three types of time-bases are maintained for the network, based on AM65xx time sync architecture.

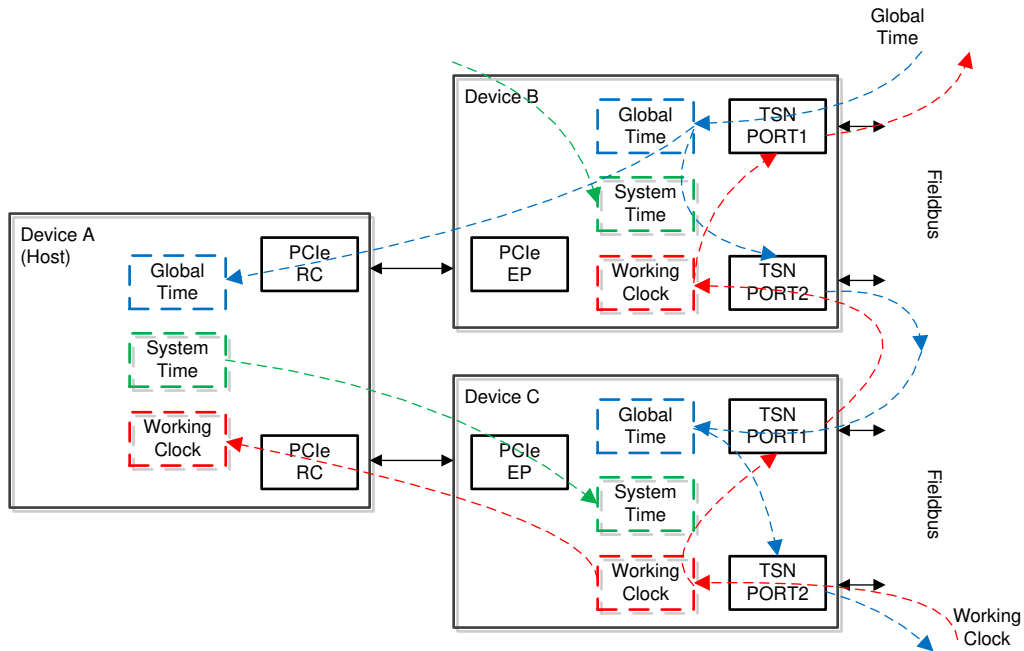


Figure 1. Example of a Control Module With Multiple Time Bases

2 AM65xx Time Sync Architecture

2.1 Functional Overview

Figure 2 shows a functional view of the time sync network in the AM65xx device, where sync signals from each protocol interfaces are interconnected via the Time Sync network.

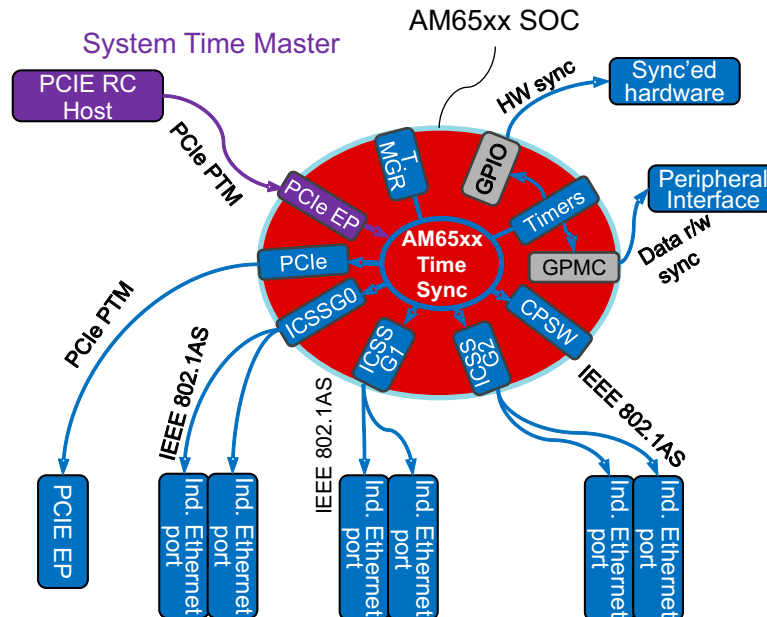


Figure 2. Multi-Protocol Time Synchronization Network in AM65xx

With this network, the following cross-protocol synchronization can be achieved:

- Time master - device can send synchronized master clock to downstream using any of these interfaces:
 - PCIe PTM Responder (Ports configured as RC)
 - Industrial Ethernet ports (IEEE 1588 or 802.1AS)
 - Ethernet port(IEEE 1588 or 802.1AS)
 - Explicit sync via hardware pins
- Time slave - receives global system time from the following interfaces
 - PCIe PTM Responder (Ports configured as RC)
 - Industrial Ethernet ports (IEEE 1588 or 802.1AS)
 - Ethernet port(IEEE 1588 or 802.1AS)
 - Explicit sync via hardware pins
- Relays global system time by receiving master time from one interface/protocol and syncing to another interface so the interface can update downstream devices
- Adjustment of on-chip Timers and Timer Managers whose timer tick can be tuned to the received global time base
- Supports hardware-based detection of clock differences between local clock and global clock, allowing CPU internal timers to use adjusted time-bases

2.2 Time Sync Components

Figure 3 shows various subsystem components of AM65xx on the time-sync interconnect. A brief description of each module is provided in the subsequent sections.

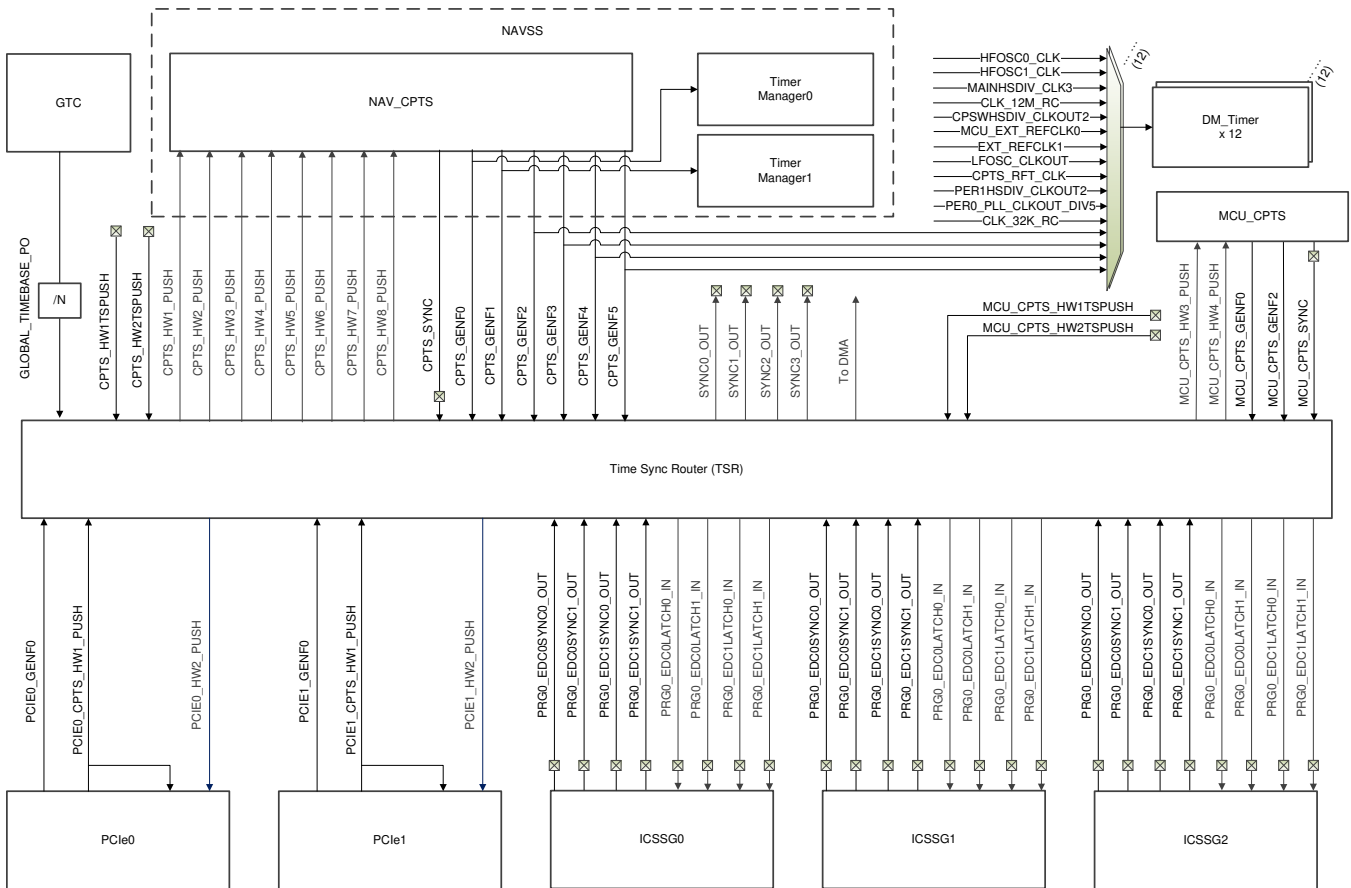


Figure 3. Time-Sync Router (TSR) and Subsystem Modules

A similar interconnect exists for the Compare Event Router (CER). For more information, see the [AM65x/DRA80xM Processors Technical Reference Manual](#).

2.2.1 TSR and CER

Time Sync Router (TSR) routes sync signals from each of the interfaces or subsystem components to one of multiple destinations. It serves as the back-plane for the time sync interconnect. There is a set of control registers that software can use to setup sync signals routing statically. CER is the similar to the TSR, but only connects timer compare events and capture events.

2.2.2 NAV_CPTS

Common Platform Timestamp (CPTS) is a hardware IP block to facilitate host control of time sync operations by collecting time sync events and present to host for processing. It performs timestamping and a comparison of generic HWPUSH (0-7) or Ethernet events (Host Transmit Event, Ethernet receive event, Ethernet transmit event, timestamp push event, timestamp rollover event, and timestamp half-rollover event). Additionally, it supports generation of “tuned” frequency waveform derived from the reference clock, where tuning may be based on parts per million, or parts per hour, with fractional register controls on the tuning rate.

Figure 4 shows a functional diagram of the CPTS block.

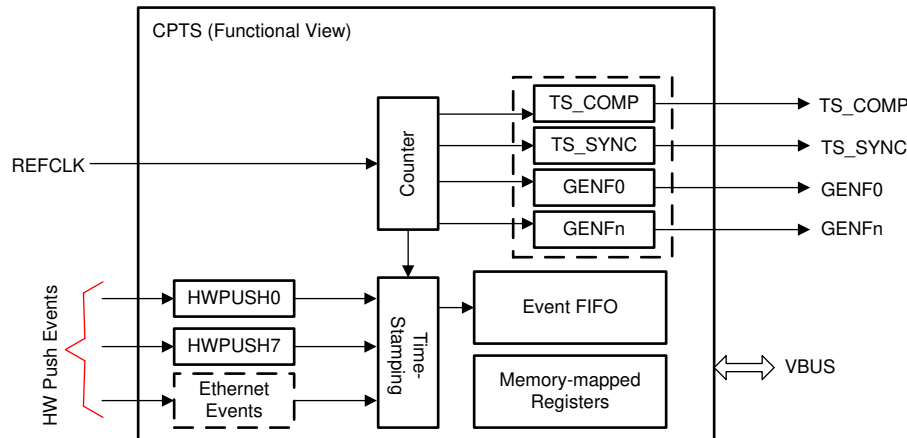


Figure 4. Functional View of CPTS Subsystem

CPTS may be integrated at the System-on-Chip (SoC) level or embedded in various subsystems where precise timestamping or event capture is needed. In the AM65xx device, there is a central CPTS in the Navigator subsystem, as well as embedded CPTS modules in each of PCIe controllers, and CPSW Ethernet controller.

2.2.3 DM_Timers and Timer Managers

DM timers are SoC-level general-purpose timers that may be used by any processor cores. Each DM timer implement a mux selector for their timer tick, four of the tick selections may be based on the GENF output of the NAV-CPTS, which enables these timers to be tuned to a master time-base received from any of the interfaces.

Timer Managers (TM) are hardware timer banks to facilitate large account of timer operations, such as activity monitoring of large amount of IO devices. Similar to DM timers, TMs can also be tuned to a master time-base received from any of the interfaces.

2.2.4 PCIe With PTM

The PCIe controllers are shown in Figure 5 support Precision Time Measurement (PTM) protocol. Figure 3 shows how the PTM module is integrated with the embedded CPTS in the PCIe controller.

In RC mode, the PTM source its master time-base from the 250 MHz pcie_txi clock, which is the pipe clock of the PCIe PHY. In EP mode, a reference clock (pcie_ptmrclk) can be selected from a set of mux input, and the reference clock is used to drive a counter in the PTM core, which subsequently drive out an adjusted output timestamps, based on PTM protocol. To directly use the timestamps as sync events for other subsystems, a selected bit of the 64-bit timestamp bus if output to the TSR. The same event is also sent to the local CPTS as a HWPUSH event so the PCIe-CPTS can directly drive sync events with additional adjustments.

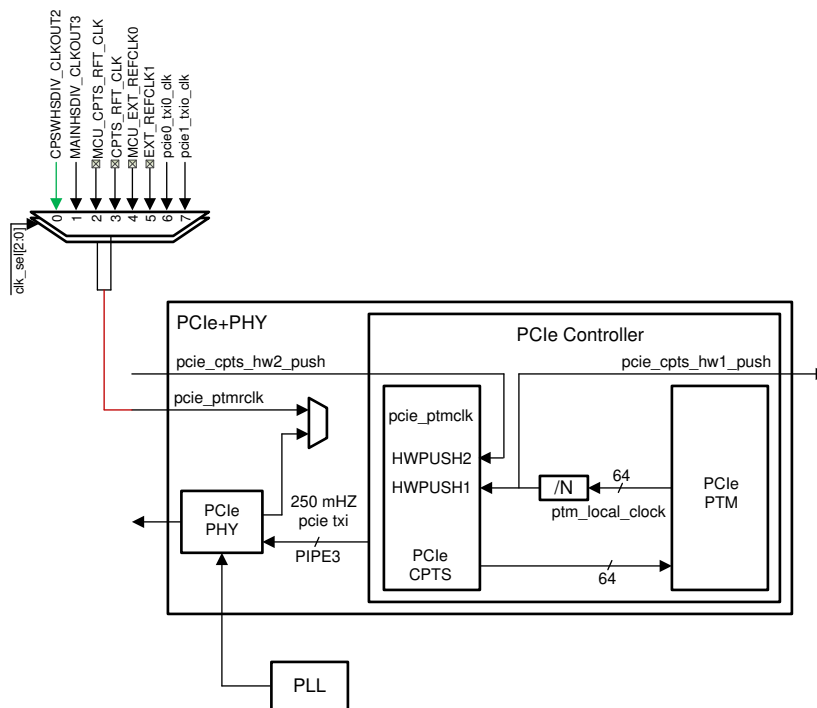


Figure 5. PCIe PTM Functional View

2.2.5 IEP Timers in ICSSGx

Figure 6 shows sync and latch signals from the IEP timers within each of the ICSSG subsystems. The two IEP timers in each of the ICSSG support 16 comparison events and 16 latch events. All comparison and latch events are connected to the CER to allow flexible event triggering and latching across the ICSSG modules. Additionally, a subset of compare events are connected to the TSR to synchronize with other modules in the device.

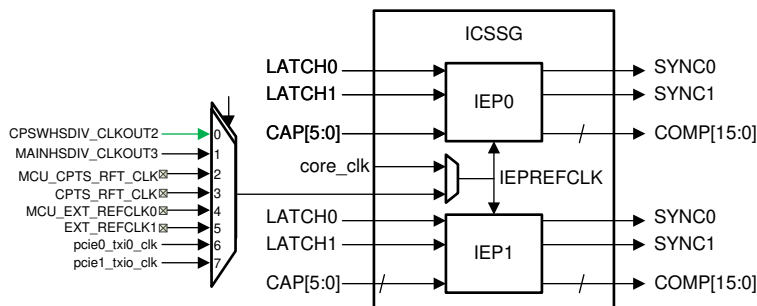


Figure 6. IEP Timers and Clock Selection

2.2.6 CPSW

The CPSW has its Ethernet events connected to its embedded CPTS, to support IEEE 1588 v2 protocols on the interface. The CPSW subsystem architecture is similar to previous devices and thus not explained in detail in this document. For more information, see the CPSW section of the device-specific TRM.

2.2.7 GTC

The Global Time-base Counter is a free running counter in the SoC, that provides Gray-coded time stamps to the A53 CPUs, where internal generic timers are derived. The GTC is a monotonic counter. There are multiple clock sources to choose for GTC's clock tick, but non of the clock source are adjustable. Therefore, A53 CPU generic timers are not tunable by hardware; instead, software adjustment should be considered if the system is required to adjust its timer based on a received master clock. [Figure 7](#) shows how GTC is integrated with the Cortex A53 CPU cluster.

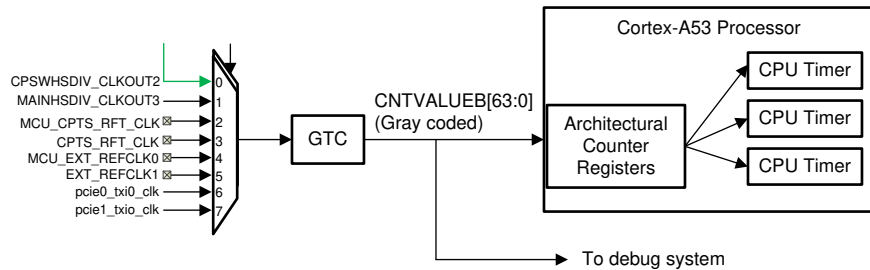


Figure 7. GTC and Clock Selection

3 Time-Synchronization Examples

In this section, two system use cases are discussed and explain how to implement these synchronization schemes on the AM65xx device.

3.1 AM65xx as the Time Master Server

In this scenario, the AM65xx device receives two master PPS signals from its timer input pins. These pins can be driven by external GPS receivers, a trusted clock source, or other types of master time-base generation units. These PPS signals are captured by NAV-CPTS module, and corresponding IEP timers are tuned to follow these master time-bases, then PTP packets are transmitted to downstream network components via IEEE 802.1AS protocol.

Figure 8 shows related clock signals and dotted lines represent software adjustment functions. In this scenario, the device maintains three time bases: system time, wall clock, and working clock.

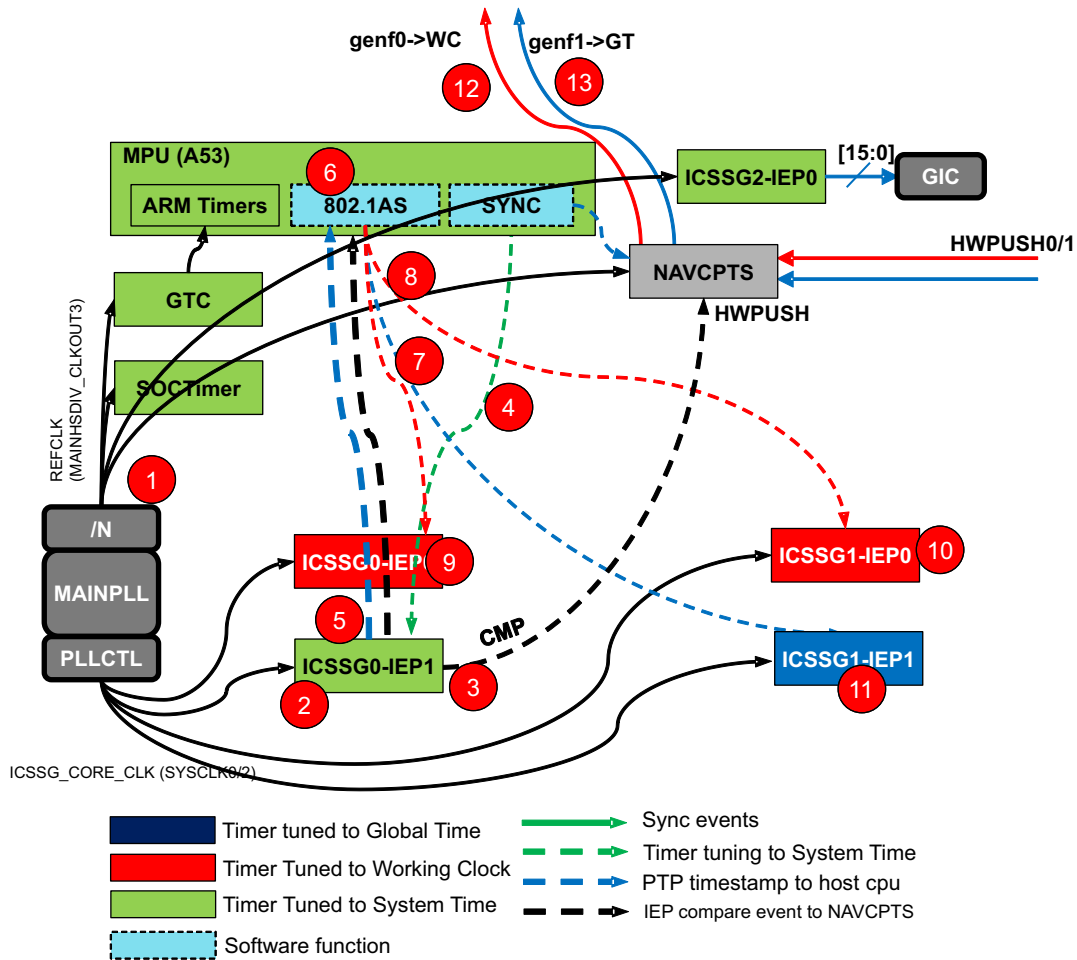


Figure 8. Time-Sync Use Case Example 1: AM65xx as Time Masters

The programming steps are as follows:

1. Configure GTC, DM Timers, NAV-CPTS, ICSSG2 all use REFCLK from MAINHS DIV_CLKOUT3. This time-base is selected as System Time.
2. Configure ICSSG0/1 to work in Sync mode, configure all IEPs in ICSSG0/1 to use the core_clk as IEP clock, running at 250 MHz.
3. Configure ICSSG0-IEP1 to generate a SYNC event, configure the event to HWPUSH of NAVCPTS.
4. The SYNC software stamps HWPUSH and calculate offset between ICSSG core clock and system time. Then, tune ICSSG0-IEP1 to System time.
5. ICSSG0 use IEP1 to timestamp Global Time and Working Clock PTP, give timestamps to the 802.1AS modules on host processor.
6. 802.1AS stack calculate actual working clock, and also
 - a. Δ_{wc} : clock offset between Working Clock and System Time
 - b. Δ_{gt} : clock offset between Global Time and System Time
7. ICSSG1-IEP1 is tuned to Global Time and used by TSN software to send out wall clock.
8. ICSSG0/1-IEP0 is tuned to Working Clock and used by TSN software to send out Working Clock.
9. Working clock is sent out by ICSSG0 for ProfiNet.
10. Working clock is sent out by ICSSG1 for TSN.

11. Global time is sent out by ICSSG1 to TSN based on ICSSG1-IEP1.
12. SYNC software configures NAV-CPTS GENF0 to send out test sync for the Working Clock.
13. SYNC software configures NAV-CPTS GENF1 and sends out the test Sync for GT. GENF1 need to be tuned to Global Time.

3.2 Multi-Domain Time Synchronization Across PCIe Interconnect

In the second example, the scenario where two AM65xx devices are deployed is considered: one as host processor and one as an interface processor, as shown in Figure 9. The interface device is connected to the host via PCIe.

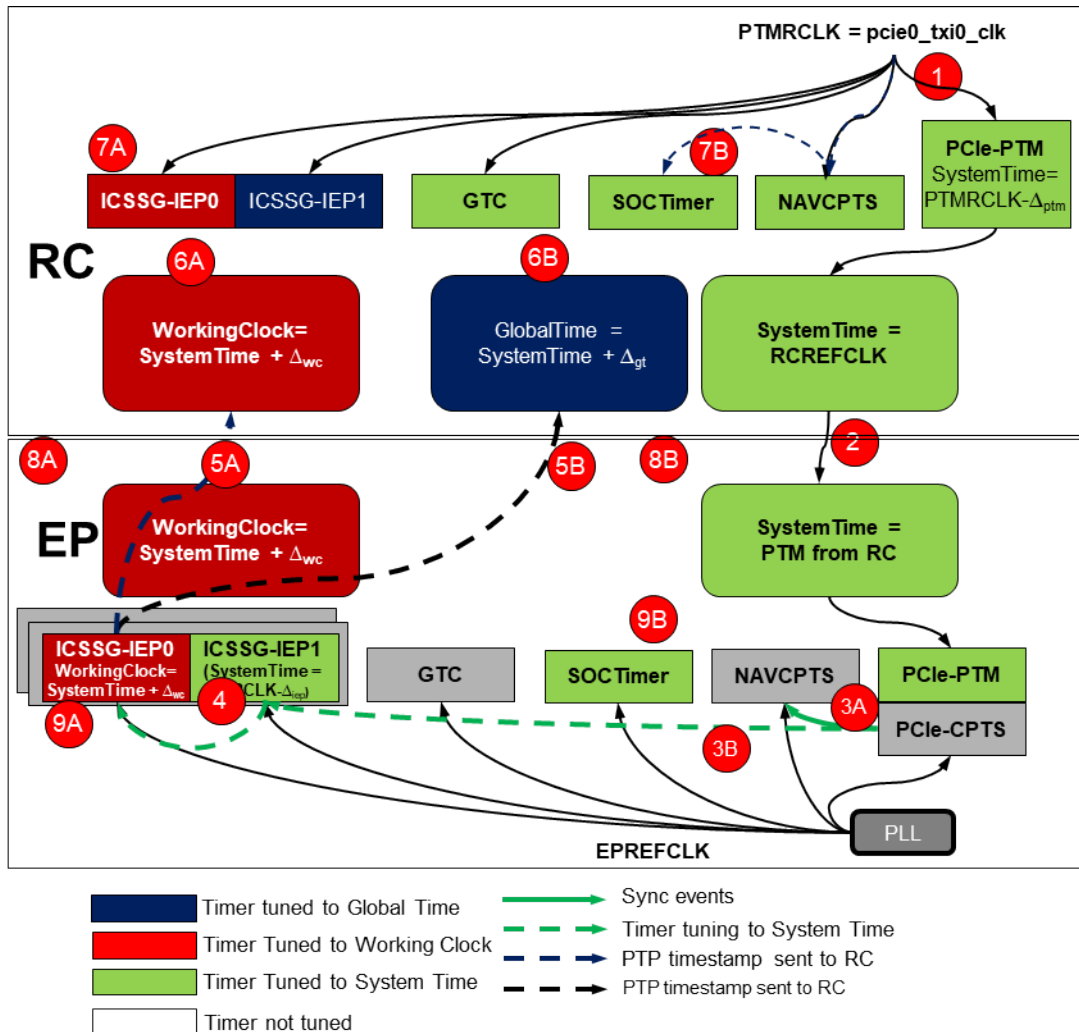


Figure 9. Time-Sync Use Case Example 2: Time Synchronization Across Two Devices

In this example, the following time-bases should be maintained:

- **System Time:** this is the commonly understood time base between the two devices, when timestamp values are exchanged across them. Typically, all timestamps shall be based on the System Time. System Time can be synchronized via the PTM protocol across the PCIe link, where the RC designate its PHY clock as the System Time. In this example, the interface device acts as PTM Requester, it then tune the IEP1 timer in the ICSSG so its firmware can timestamp PTP packets based on the common System Time;
- **Working Clock:** this is sometimes referred as the communication time, it is the common time-base for network packet scheduling and traffic management. In this example, the device receives 802.1AS PTP streams from its network interface, then ICSSG firmware parses PTP streams and send out recorded timestamps to the host for 802.1AS protocol execution.
- **Global Time:** this is the time all time sensitive tasks obey in the domain. Typically these time-sensitive applications are scheduled based on the Global Time. In this example, similar to the Working Clock, the Global Time is received on the Ethernet port via the IEEE 802.1AS protocol. The ICSSG firmware parses PTP packets and send timestamps to host where the actual protocol stack runs.

Some detailed configuration and software steps are:

- RC designate PTMRCLK = pcie0_txi0_clk as System Time and configure other timer modules to use this clock as timer tick
- RC sends system time to EP via PTM protocol
- EP receives updated PTM time, sync events are generated from PCIE to trigger time-stamping by both NAVSS-CPTS and ICSSG-IEP1. Rate and phase offset between EP-REFCLK and PTM: Δ_{ptm} is recorded
- ICSSG-IEP1 is tuned by applying Δ_{ptm} , then the tuned IEP1 is used to timestamp PTPs for Working Clock (WC) and Global Time (GT), and record:
 - T_{wc} : WC PTP stamped by System Time, there may be multiple timestamps according to 802.1AS
 - T_{gt} : GT PTP stamped by System Time, there may be multiple timestamps according to 802.1AS
- T_{wc} and T_{gt} are sent to RC by EP via PCIe interface (not part of PTM protocol);
- RC execute 802.1AS protocols to calculate working clock and global time, and also calculate:
 - Δ_{wc} : clock offset between Working Clock and System Time
 - Δ_{gt} : clock offset between Global Time and System Time
- RC tunes ICSSG-IEP0 for working clocking and tunes one of SoC Timers for Global Time
- RC sends Δ_{wc} and Δ_{gt} to EP via PCIe interface (software).
- At EP side, ICSSG0-IEP0 and one of the SOC timers are tuned to the WC and GT, respectively

3.3 Hand-Over and Recovery

In real systems, it is important to have secondary time masters or hold-over mechanisms when the primary time master is not available, or when the communication channel is broken. Since AM64xx time-sync architecture allows delta adjustment to a local clock source, it is natural to maintain the same adjustment values during the period when no new adjustment values are available.

In case of loss of primary time master, the AM65xx device can be designated as the backup time master, where it can seamlessly resume the role of distributing master time bases. When the primary time master resumes, the secondary time master can hand-off the mastership to the primary. Detailed handoff mechanisms should be designed to ensure that certain time bases do not roll back.

4 Summary

While protocols are well defined for communication and network protocols, it is equally important to have hardware support within the SoC, to achieve designed time sync precision. It is critical to have the required time-stamping hardware as well as time sync interconnect in the device.

With the deployment of Time Sensitive Networks (TSN) in factory automation, industrial controls, automotive, and rest of Industry 4.0 networks, it became essential for a processor to support multiple time-sync protocols simultaneously, maintaining multiple time domains in the device, as well as cross-protocol synchronization.

The capabilities of the time sync architecture expands far beyond the two example use cases explained in this document. Actual system implementations may implement more flexible synchronization, hand-off mechanisms and robust recovery functions when time masters become non-available.

5 References

- Texas Instruments: [AM65x/DRA80xM Processors Technical Reference Manual](#)

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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, 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 (www.ti.com/legal/termsofsale.html) 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.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated