Intel® Time Coordinated Computing (TCC) User Guide

(Public-Version[1])

January 2026

Notices & Disclaimers
Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.
Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.
Your costs and results may vary.
Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.
Intel® Turbo Boost Technology requires a PC with a processor with Intel Turbo Boost Technology capability. Intel Turbo Boost Technology performance varies depending on hardware, software and overall system configuration. Check with your PC manufacturer on whether your system delivers Intel Turbo Boost Technology. For more information, see http://www.intel.com/technology/turboboost.
© Intel Corporation. Intel, the Intel logo, Intel Core, Intel SpeedStep, Intel vPro, VTune, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

[1] An extended version of this guide with additional information regarding Intel's Time Coordinated Computing offerings is available for customers under NDA at RDC #786715

Document Number: 831067

Figures

Tables

Revision History

Date	Revision	Description
August 2023	1.0	Initial External Release
September '23	1.1	Updated with Precision Time Coordination
February'24	1.2	Updated with 14th Gen Intel® Core™ S-Series Processor information and additional content updates
March'24	1.3	Added Intel Atom® X7000RE/C Series processor (Amston Lake) data
May '24	2.0	Added TGPIO pin info, cyclic test command line info, additional information on power management settings and several advanced tuning options; Addition of Intel® Xeo® D-2896TER KPI data
June'24	2.1	Added enveloping usecase for Intel® Speed Shift Technology for edge compute on 13th Generation Intel® Core™ H-Series processor
August'24	2.2	Added additional 14th Generation intel® Core™ SKUs with Intel® TCC support
September'24	2.3	Added Intel® Ethernet Controller i226 AVNU Component Certification and Intel® Speed Shift Technology for edge compute for 13th Generation Intel® Core™ H-Series and P-Series processor
March'25	2.4	Editorial changes describing Intel offerings supporting the TCC experience. Added new references and software resources.
January'26	2.5	Various editorial enhancements, more details on real-time capabilities and new references.

§

1.0 Introduction

A major transformation towards software-defined solutions and AI is happening at the Edge. It's driven by factors such as the need for flexibility and scalability, changes in the workforce, advances in Edge-based AI and more. A key element of this transformation is the capability to support real-time workloads along with best-effort workloads on the same system. Moreover, emerging software-defined systems and AI applications are distributed, involving multiple computing nodes (e.g. multiple sensors, controllers, actuators) that perform coordinated actions based on a shared understanding of time. This distributed computing paradigm, known as Time Coordinated Computing, is crucial across Edge applications including real-time systems, industrial automation, distributed displays, large-scale simulations and digital twins leveraging Edge AI.

Intel Platforms include optimized hardware capabilities and a strong software ecosystem to support mixed-criticality workloads and ensure real-time performance delivering a great Time Coordinated Computing experience.
These optimizations are designed to make Intel platforms with real-time support achieve high determinism in the presence of best effort workloads running on the same system.
Intel's Time Coordinated Computing offerings provide the following key values:

- Timely & reliable data processing: Better out-of-box real-time performance for deterministic workloads
- Doing more with the same system: Maximum efficiencies by aggregating real-time and best-effort applications on a single system
- Timely & reliable data delivery: Support for IEEE Time Sensitive Networking (TSN) over converged networks

- Future-proofing designs: Scale between Intel Atom®/Entry processors, Intel® Core™ and Core™ Ultra processors, and Intel® Xeon® processors as well as across processor generations.

This document describes real-time features and optimizations across several processor families and generations. While real-time optimizations are available across more processors in every new generation, some features may not be available on all product stock keeping units (SKUs) and product families. Contact your Intel representative for details.

Real-time applications (also called “workloads”) must execute within a certain amount of time, consistently, across numerous iterations. While specific requirements vary by use case, real-time applications typically perform the following sequence of tasks:

1. Process new input such as a sensor measurement or camera video streams.
2. Perform a computation such as a new motion vector or object detection bounding box of a computer vision workload.
3. Control an actuator or render a result.
4. Ensure actions are tightly synchronized or happen at the exact same time.

Furthermore, these applications often have a deadline. They must complete the tasks within a certain time, which may range in typical applications from microseconds to milliseconds.

Intel provides the following support for real-time applications and a great Time Coordinated Computing experience:

- Selected processors with features to minimize worst-case execution time (WCET) within the system.
- Selected processors with time synchronization optimizations.
- Ethernet controllers and wireless connectivity solutions that support IEEE 802.1 Time-Sensitive Networking (TSN) standards.

- Validated configuration guidance as the base for Intel's key performance indicators (KPIs) of real-time performance.
- Reference software to help development of real-time systems using Intel processors and connectivity solutions.

1.1 About This Document

This document provides guidance on how to utilize Intel's platforms for real-time usages, including some technology background as well as recommendations and examples on how to prepare Intel's real-time capable edge platforms for use with real-time applications. This document is for anyone working on these components, such as:

- System engineers preparing a system to support real-time applications.
- Engineers at ODMs and OEMs setting up systems and preparing the base platform software as well as validating its readiness for real-time.
- Software developers developing and validating SW applications and middleware with real-time requirements to work on Intel systems.

Consider this document the starting point for understanding how to best use Intel platforms for real-time usages. This document:

- Introduces the basics of running real-time workloads on Intel systems.
- Provides recommendations and examples on how to achieve the desired real-time performance on Intel platforms.
- Introduces the KPIs that Intel uses to measure real-time performance.
- Introduces real-time hardware features and related software.
- Provides information for advanced tuning of Intel platforms for users with very stringent real-time requirements.

A more detailed NDA-only version of this guide is also available (RDC#786715).

Note: This document is not intended to provide a comprehensive specification for real-time features or software implementations. Where feasible, reference documentation has been cited that gives further technical detail for the topics discussed.

2.0 Real-time Capabilities Overview

2.1 Hardware Features

Intel® Core™ and Core™ Ultra Processors, Intel Atom®/Entry Processors and Intel® Xeon® Processors provide hardware features to optimize real-time compute performance and support ethernet controllers with IEEE TSN capabilities to optimize performance in converged networks.

In this chapter, the hardware features are described at a high level. For information about hardware features beyond the scope of this document, see the External Design Specification (*EDS) documentation* for the specific platform in RDC.

2.1.1 Hardware Optimizations Overview

Intel's offering comprises many optimizations throughout Intel's processors and connectivity solutions targeted at improving real-time performance in a mixed criticality environments where real-time as well as best-effort workloads run concurrently and utilizing the same network resources.

2.1.1.1 Time Synchronization Optimizations

- Precision Time Coordination (PTC): Clocks in the SoC subsystem are precisely correlated via hardware time-stamping.
- Timed General-purpose I/O (TGPIO): General purpose I/O pins that can be used to either import time to or export time from the platform.
- PCI Express* Precision Time Measurement (PTM): Enables precise coordination of real-time events across multiple PCIe connected components with independent local time clocks.
- IEEE 1588 (PTP) and the IEEE 802.1AS TSN profile of IEEE 1588 are supported in Intel end-point connectivity solutions.

2.1.1.2 Timeliness Optimizations

- Power State Transition Optimizations: Enable CPU to keep executing instructions while its frequency is increasing or decreasing. While on early platforms supporting TCC, e.g. Intel Atom® x6000 Series Processors and 11th Generation Intel® Core™ Processors), Intel recommended disabling many power management capabilities, subsequent additional improvements have changed such recommendations to allow some power management features to remain enabled with minimal impact to real-time behavior. In addition, Intel® Speed Shift technology together with other P-State options may be used to boost real-time performance by increasing the core frequency on select cores.
- Memory/Cache Allocation Optimizations: Partitioning of shared caches at the way level between classes of service (L2 & L3 Cache) - Intel® Cache Allocation Technology (CAT); Limit amount of cache available to GPU (GT-CLOS).
- Interrupt Request (IRQ) Optimizations: Optimize processor microcode & other overhead in the critical path for interrupts in the CPU core; Allow devices to deliver interrupts directly to the guest OS without requiring preprocessing by the hypervisor.
- Fabric and PCIe Virtual Channels: Virtual channels on Fabric and PCIe available to high priority workloads.
- Intel® Speed Shift for Edge Compute Applications enables specific assignment of processor performance to where it is most needed.
- Support of IEEE TSN standards (e.g. 801.1AS, 802.1Qbv, 802.1Qav, 802.1Qbu/3br) in end-point connectivity solutions to take advantage of TSN features becoming common in edge networks.

2.2 Supported Hardware

Intel offers real-time support for Time Coordinated Computing on a wide variety of platforms and SKUs. It includes a set of optimizations in the SOC hardware and processor microcode including support for L2 and LLC (L3) Cache Allocation Technology (CAT) as well as IEEE Time Sensitive Networking (TSN) support in our connectivity offering.

2.2.1 Intel processors supporting Real-Time and Time Coordinated Computing Experience

A wide variety of Intel processors support real-time enhancements and support for the Time Coordinated Computing Experience including many versions and SKUs of Intel® Xeon® processors, Intel ® Core™, Intel Core™ Ultra processors, and Intel Atom® processors. Appendix A provides a detailed list of processors and SKUs offering real-time support.

Note: For the latest details on specific SKUs and platform capabilities, please refer to the information provided at: https://www.intel.com/content/www/us/en/products/details/embedded-processors.html.

2.2.1.1 Intel® Xeon®

Intel® Xeon® processors carry robust system partitioning capabilities under the resource director technologies. For the best real-time performance it is recommended to utilize these capabilities see the Intel® Resource Director Technology (Intel® RDT) Reference Manual for more details.

2.2.1.2 Intel® Core™ and Core™ Ultra

Intel® Core™ and Core™ Ultra processors generally support a subset of the partitioning capabilities found in Xeon, some of these capabilities are only partially implemented, for example cache allocation technology is supported but not architectural, as described in Section 3.6. These capabilities can still be used to support the most demanding real-time use cases, see Section 3.6 for more details.

2.2.1.3 Intel® Atom®

Intel® Atom® and other entry processors historically have targeted embedded use cases, many with real-time constraints, as a result the Atom line of processors have implemented many features to enable robust real-time performance even while under load. This guide carries details for the simple configuration of these capabilities.

2.2.2 TSN-capable Ethernet Controllers

Intel platforms with real-time support feature TSN over Ethernet via Intel's discrete i226-LM/IT Ethernet Controllers and on select processor SKUs and product families via integrated Ethernet MACs (for Intel® Core™ S-Series processors the R680E SKU of PCH is required for integrated 2.5Gb Ethernet with TSN support)[1].

In September 2024 the Intel® Ethernet Controller i226 became the first endpoint device granted Avnu Alliance Component Certification for compliance with the IEEE 802.1AS (Timing and Synchronization for Time-sensitive Applications) and IEEE 802.1Qbv (Enhancements for Scheduled Traffic) standards under the IEEE TSN toolbox.

Table 1: TSN-capable Ethernet Controllers

Ethernet solution	Number of Ports	Media-Access Control (MAC)	Base-T PHY	Supported on
Discrete Intel® Ethernet Controller i226 LM/IT (2.5GbE with TSN and vPro support)	1 port per i226	Discrete PCIe device (MAC/PHY combo)		All TCC supported processors Avnu Alliance Component Certified for compliance with the IEEE 802.1AS and IEEE 802.1Qbv [Avnu Certified Component Registry]()
Integrated 2.5GbE MAC[2]	1 port	Yes	3rd Party	• 11th Generation Intel® Core™ UP3-Series Processors • 13th Generation Intel® Core™ U-, P-, H- Series Processors • 14th Generation Intel® Core™ S-Series Processor • 15th Generation Intel® Core™ Series 2 Processor • Intel Atom® x7000E series processors
Integrated 2.5GbE MACs	2 ports	Yes (1 per port)	3rd Party	• 13th Generation Intel® Core™ S-Series Processors (R680E SKU of PCH only) • 12th Generation Intel® Core™ S-Series Processors (R680E SKU of PCH only)

[1] TSN capabilities were supported with Integrated Ethernet MAC in Intel platforms launched up to 2025. Future platforms will only support TSN on discrete Intel Ethernet Controllers.

[2] The Integrated 2.5GbE MAC has a known issue to perform 2.5G PHY auto-negotiation when used with 13th Gen Intel® Core™ processors.

Ethernet solution	Number of Ports	Media-Access Control (MAC)	Base-T PHY	Supported on
Integrated 2.5GbE MACs	3 ports	Yes (1 per port)	3rd Party	• Intel® Xeon® W-11000E Series Processors • Intel Atom® x6000E series processors

The ethernet controllers with TSN support can operate at multiple speeds: 10Mbps, 100Mbps, 1Gbps, and 2.5Gbps and in either full duplex or half duplex mode. In upcoming Intel platforms, TSN capabilities will be supported only by discrete Intel Ethernet Controllers.

The integrated ethernet MAC with TSN support is accessed by Intel's processor cores through system software as a PCI express Root Complex Integrated Endpoint (RCiEP) via PCH I/O Fabric (PSF2) and only supports the SGMII interface.

To support Ethernet with TSN, the integrated ethernet MACs require 3rd party Ethernet PHYs external to the SOC. Supported PHYs are generally listed in the product briefs of the processor families.
Note: Many other Intel® Ethernet Controllers support IEEE 1588 and PTM (see product specifications for details).

2.2.3 TSN-capable Wireless Solutions

Intel is working on providing TSN support over 5G and Wi-Fi on select Intel platforms. This section will be updated as such TSN capabilities become available.

2.3 Software Capabilities

Intel provides software that enables developers to access hardware optimizations for real-time performance, including BIOS support, Linux* kernel driver and user-space patches, and sample applications. In addition, Intel is working with ecosystem partners to deliver TCC and IEEE TSN support as part of 3rd party SW solutions.

2.3.1 Operating System Support

Linux OS

Intel provides open-source drivers and user-space interfaces in Linux to enable real-time capabilities in Intel's products. Linux also supports real-time performance in the mainline kernel by integrating the PREEMPT_RT model, which enables deterministic scheduling and low-latency response for real-time workloads. For proper configuration of Linux for real-time performance, see Section 3.4.

Intel also delivers new real-time features and bugfixes on the latest upstream Linux kernel versions, as found at: https://kernel.org/

Intel maintains a set of features and bugfix kernel backports targeting the current and previous upstream LTS versions. These overlays are distribution-agnostic and can be downloaded from:

https://github.com/intel/linux-kernel-overlay/

Some of the new kernel features may require changes in user space interfaces and tools. In these cases, Intel contributes the changes to upstream projects, and rely on open-source distributions and commercial OSV partners for their distribution.

Other Operation Systems

Intel works with select real-time operating system vendors (e.g., Acontis, IntervalZero, QNX, SYSGO, TenAsys, and WindRiver) and contributes to open-source RTOSs (e.g. Zephyr, Xenomai) to enable support for Intel real-time and TSN capabilities.

2.3.2 Hypervisor Support

Intel enables 3rd party Hypervisors for real-time support, such as the Real-Time Hypervisor from Real Time Systems (RTS) and validates select capabilities with open source Hypervisors such as ACRN and KVM .

2.3.3 UEFI-BIOS Support

Many of the control registers involved in configuring a platform for real-time usage are accessible through the BIOS.

Intel's reference BIOS includes various real-time enhancements including an option called Time Coordinated Computing Mode (TCC Mode), a single BIOS switch that optimizes the firmware settings for low latency. If your BIOS vendor has included support for TCC Mode, enabling this setting is a quick way to optimize your firmware configuration for real-time performance. See Section 3.3.1 for more details on the TCC Mode.

2.3.4 Slim Bootloader Support

Like the UEFI-BIOS, Intel's Slim Bootloader (SBL) reference provides a configuration option to enable or disable TCC Mode.

2.3.5 Reference Software Resources

Intel works with the broader community to enable a strong SW ecosystem to take advantage of platform optimizations and support a great Time Coordinated Computing experience. The following resources are available to help developers and users building applications that need real-time performance:

- **TCC Tutorial**: A step-by-step guide on optimizing Linux Real-Time performance on Intel CPUs available at https://github.com/intel/edge-developer-kit-reference-scripts/tree/main/usecases/real-time/tcc_tutorial

- **Canonical's CTO guide to RT Linux**: recommended reading as an introduction to Linux real-time capabilities. Available at https://ubuntu.com/engage/cto-guide-real-time-kernel

- **Linux Real-Time Communications Testbench**: an open-source tool for validating and testing real-time applications and TSN capabilities on Linux systems. Available at https://github.com/Linutronix/RTC-Testbench

- **TSN reference applications**: an Intel developed set of TSN sample/reference applications available at https://github.com/intel/iotg_tsn_ref_sw.

Clock Manager: A software tool for monitoring network time synchronization on the local platform, supporting single and multiple time domains. Available at https://github.com/erezgeva/libptpmgmt/tree/master/clkmgr

3.0 Time Coordinated Computing Quick Start Guide

3.1 Getting started with Intel® TCC

As mentioned in Chapter 2, Intel platforms are designed to support various applications and workloads and serve a broad range of markets. Intel's real-time offerings include several HW and SW capabilities to enable workloads requiring real-time performance (low latencies, deterministic processing, and accurate time synchronization) in mixed-criticality environments.

Many elements influence the real-time performance of a platform including HW and SW choices, BIOS and SW settings as well as ways to configure shared resources on the platform. As described in the previous chapters, Intel platforms include many real-time optimizations in hardware and software. The level of determinism required strongly depends on the real-time workload. Hence, not every implementation will require all the HW and SW optimizations available on Intel platforms to be enabled. This chapter describes some of the basic considerations for running real-time workloads on Intel platforms.

The following table provides guidance on how to approach usage of the capabilities offered.

Table 2: Recommendations for achieving real-time performance with Intel platforms.

1	Choose an Intel platform with real-time (including TCC and TSN) support	This ensures your platform has all the needed capabilities to support real-time workloads. This generally also includes choosing a connectivity solution with IEEE TSN support.
2	SW stack selection	Select the best RTOS and Hypervisor to fit your needs (e.g., Linux Preempt-RT)
3	Apply real-time kernel boot parameters	Follow the directions for low latency configuration of your OS by your OS vendor (kernel boot parameters)
4	Pin your real-time workload to cores	Pinning a real-time workload to a core takes away some real-time performance variance. In a Hybrid system pinning allows allocating the real-time workload to the type of core best fitting the needs of the workload.
5	Configure platform for Time Coordinated Computing via BIOS, providing the most flexibility for mixed criticality applications	Evaluate the overall performance of your system as well as your ability to support your real-time workloads after enabling **TCC Mode** in your BIOS. If your system does not support TCC Mode, please refer to alternate BIOS configurations detailed in Appendix B. For many applications enabling TCC Mode or equivalent BIOS settings will likely be sufficient!
6	**If needed:** Allocate L2/3 cache to high priority workloads	Use Intel® Cache Allocation Technology to allocate an amount of cache for the real-time workload to utilize and also take into consideration other cache accessing agents, such as integrated graphics.
7	**If needed:** Configure Intel® Speed Shift to increase real-time performance for edge compute applications	Use Intel® Speed Shift technology to boost core frequency on cores with high priority real-time workloads.

3.2 Hardware Configuration

Intel processors are designed to be configurable to address many unique customer requirements. Such requirements can be form factor (layout), thermal, performance, and cost. The combination of these requirement categories will ultimately determine board characteristics like memory type & speed, memory layout, and I/O layout. Regardless, for best real-time performance, Intel recommends:

- Populating at least 1 DIMM per channel for the maximum number of channels supported
- In general, both CPU die and PCH PCI Express slots can be used for real-time workloads and support optimizations such as PTM. Utilizing PCI Express slots attached to CPU die provides the lowest latency / highest throughput network connections.

Note: Refer to the documentation provided by the board or system vendor to identify which PCI Express connectors route to the CPU die.

3.3 Firmware Configuration

Intel's reference UEFI and Slim Boot Loader (SBL) include multiple knobs to configure the firmware for real-time performance. Such knobs include enabling and disabling specific functionalities that tend to impact real-time performance, such as some power-, frequency-, and thermal-management features, hyperthreading and capabilities like PCIe Precision Time Measurement. As described next, there is a global knob (TCC Mode) which sets default values for all knobs considered relevant for real-time performance.

3.3.1 TCC Mode

TCC mode includes a wide range of platform specific real-time optimizations including the following:

- Power states and frequency transition optimizations
- Configuration of real-time supporting features, e.g.:
    - Limit amount of cache available to GPU.
    - Optimize IO device utilization of Cache (for select platforms).
    - Set up virtual channels for VC-capable endpoints.

For UEFI, the TCC Mode option is generally found in the following firmware configuration menu, but specific naming may vary in your menu:

Intel Advanced Menu > Intel® Time Coordinated Computing

For Slim Boot Loader (SBL), this configuration option is found in the following configuration file in the SBL source directory: Platform\CommonBoardPkg\CfgData\CfgData_Tcc.yaml
You can override the enable or disable TCC Mode using the ConfigEditor tool. For more information on updating the settings using this tool, refer to the SBL wiki.

If your UEFI, or SBL vendor has included support for TCC Mode, Intel recommends enabling this setting as a starting point for optimizing the firmware settings for real-time performance.

TCC Mode configures various existing UEFI, or SBL settings to predetermined values. The settings altered by TCC mode can be set independently for granular control of specific capabilities whether TCC mode is enabled[1] or not. We generally recommend not deviating from the default Intel TCC Mode values to ensure consistent configuration that is optimized for general real-time performance.

In cases where your firmware configuration does not include support for TCC Mode, please see Appendix B for platform specific settings that can be used to manually implement the equivalent TCC Mode functionality.

Note: Some firmware configuration options may be overridden by the operating system. Specifically, when using Linux, the relevant kernel parameters should be supplied during boot to prevent the OS from altering the firmware configuration settings.

Note: Altering hardware and software power management features can negatively affect real-time performance and general compute performance for various I/O paths.

---

[1] For older platforms, like Intel Atom® X6000 series processors and 11th Generation Intel® Core™ processers, some settings set by TCC mode cannot be altered once TCC mode is enabled.

3.3.2 Power Management settings

As mentioned in Chapter 2, Intel has optimized power management capabilities on recent platforms resulting in improvements of real-time performance even with some power management enabled. Starting with the 12th Generation Intel® Core™ processors and the Intel Atom® x7000 Series processors, TCC Mode no longer changes the default settings of Intel® Speed Shift Technology and Intel® Speed Step Technology. In addition, we keep Intel® Turbo Boost technology enabled during our real-time performance measurements on those platforms, as highlighted in Chapter 4: Key Performance Indicator Measurements.

3.4 Operating System Configuration

The operating system is a critical element for real-time optimization and options depend heavily on the particular operating system used. Please check with the OS vendor on the available options.

The Linux kernel is capable of real-time performance when properly compiled and tuned.

The PREEMPT_RT preemption model must be selected when configuring the compilation. Since version 6.12, this is available in regular mainline kernels. When using previous versions, a patch has to be applied.

Recent Linux distributions may provide a kernel pre-compiled with the PREEMPT_RT preemption model. Those packages usually include "-rt" in their name.

Please refer to each distribution for further instructions about how to install and tune Linux for real-time performance:

- Canonical Real-time Ubuntu*[1]: Tuning a real-time kernel
- RedHat RHEL*: Understanding RHEL for Real-time
- SuSE*: SUSE Linux Enterprise Real Time

[1] Other names and brands may be claimed as the property of others.

As an example of real-time tuning using the kernel command line parameters, the following could be used to ensure the timestamp counter in the CPU is used and interrupts to the cores reserved for real-time are avoided:

clocksource=tsc
tsc=reliable
nmi_watchdog=<BE cores>
nosoftlockup
idle=poll
isolcpus[1]=<RT cores>
rcu_nocbs=< RT cores>
nohz_full= < RT cores>
irqaffinity=<BE cores>

where <RT cores> is a cpu list of one or several cores dedicated to real-time tasks disjoint to the list <BE cores> of best effort or house keeping cores which might be e.g. core 0.

3.5 Optimal Clock Synchronization Accuracy

Intel platforms support Precision Time Coordination (PTC) out of the box, enabling accurate intra-platform clock synchronization between various subsystems of the SOC. PTC is an Integrated On-chip System Fabric (IOSF) sideband-based protocol that allows IPs to sync and resync at any time that any IP wishes to do so, providing a system-wide basis for precision time.

Precision Time Measurement (PTM), an optional feature introduced in the PCI Express 3.0 standards, synchronizes the host system clock with the network adapter's PTP clock. PTM acts as a critical bridge that extends network time precision to the host system, enabling applications to access network-synchronized time with minimal accuracy degradation. I/O devices supporting PTM deliver the highest level of clock synchronization accuracy.

PTM employs a hardware-based cross-timestamping mechanism to correlate clocks throughout the platform - specifically between the PCIe root complex (such as the host CPU) and connected endpoints

[1] While isolcpus use is not generally advised as it cannot be changed at run-time, for real-time applications it can be used for CPU isolation.

(like network adapters). This synchronization is essential because applications depend on the system clock for timing operations. Without PTM, software-based access to the network adapter's time introduces substantial jitter ranging from hundreds of nanoseconds to microseconds, effectively negating the precision benefits that PTP provides.

To verify time synchronization accuracy at the application level (including the functionality of PTM), Timed General-Purpose I/O (TGPIO) pins are available on the CPU subsystem. These I/O pins are linked to the host CPU's hardware clock, enabling both the validation of time at the host as well as the extension of the unified time domain to external hardware. Further information on TGPIO can be found in Chapter 5.

3.5.1 Enabling PCI Express Precision Time Measurement (PTM)

PCIe PTM is supported on Intel Atom® x6000 Series and later platforms, Intel® Core™ 11th Generation and later platforms, and Intel® Xeon® 4th Generation and later platforms. PTM functionally can generally be enabled/disabled in the firmware configuration. While PTM functionality is typically enabled by default, if it is not already enabled, there are two ways to enable PTM:

1) If your firmware configuration supports TCC Mode, the easiest way is to enable TCC mode as described in Chapter 3.3. TCC mode will automatically enable PTM on all PCIe root ports on the processor.
2) However, if your firmware configuration does not support TCC Mode, you can verify/enable PTM for the platform. On Intel's reference, the UEFI Menu option to enable PTM per port can generally be found at **Intel Advanced > System Agent (SA) Configuration > PCI Express Configuration** and **Intel Advanced > PCH-IO Configuration > PCI Express Configuration.** The action needs to be repeated for any PCIe root port the user wishes to enable.

To take advantage of PTM, all devices in the path from the root port to the PCIe endpoint must support PTM. For example, if a PCIe switch connects several Ethernet controllers to a single PCIe root port on the platform, both the PCIe switch and the Ethernet devices must support PTM.

3.5.2 Supporting Multiple Time Domains

Some systems/applications may require access to multiple clocks. For instance, to enable 2 application clocks, the software and platform must support:

- 2 PTP domains
- 2 GM instances connected to the network (provisioning and maintenance of the GM are outside of the scope of the platform)
- 2 PTP4L instances (one per PTP domain)
- 2 vClock instances (one per PTP domain)
- 2 System Time Synchronization Daemon (e.g., Chrony) instances (one per application clock).

Between each GM instance and its corresponding application clock, virtual clocks (vClocks) are used for packet timestamping. Virtual clock support was finalized in the 5.19 kernel, enabling support for multiple PTP time-bases without additional hardware by transforming the hardware timestamp clock in software. The ABI for vClocks is documented at https://www.kernel.org/doc/Documentation/ABI/testing/sysfs-ptp. These timestamps are used by PTP4L to compute the offset between the GM and the local device clock. Support for multiple domains using PTP4L and the IEEE 802.1AS protocol requires support for Common Mean Link Delay Service (CMLDS), which requires version 4.3 or newer of Linux PTP (https://linuxptp.sourceforge.net/). A virtual clock (vClock) is disciplined by PTP4L to track the GM instance.

An application clock (e.g., aka working clock and global clocks) is disciplined to track a vClock using Chrony (https://chrony-project.org/download.html). The ABI for application clocks is documented at https://www.kernel.org/doc/Documentation/ABI/stable/sysfs-kernel-time-aux-clocks. Each time-base is mapped from its GM to a clock that is directly useable by the application through a call to POSIX clock_gettime(). An example of this flow is found in the image below.

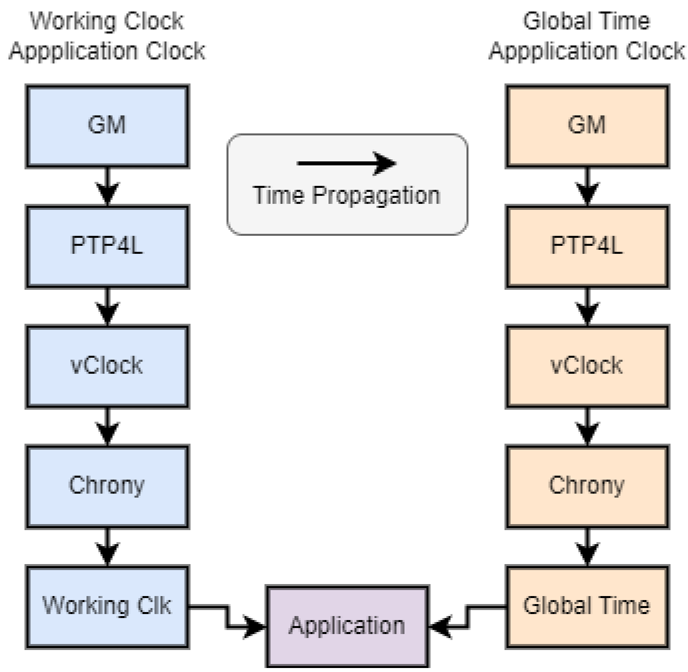

Figure 1: Multiple clock synchronization blocks and flow.

3.6 Cache Allocation for Real-time Workloads

In real-time system designs, it is often necessary to have control over shared resources to ensure that an adequate amount of that resource is available when needed by the application. Failure to allocate sufficient resources can lead to a performance degradation of the real-time application, causing it to miss execution deadlines.

System caches are one of the shared resources that, depending on the amount of contention, can drastically affect the performance of a real-time application. Intel provides the ability to partition cache resources using Cache Allocation Technology (CAT), which, depending on processor family and generation, may be available on both the level 2 (L2) Cache and Last Level Cache (LLC). Both L2- and LLC-CAT are available on Intel processors supporting real-time performance, however on Intel® Core™ processors and Intel Atom® processors, LLC (L3 cache) is non-architectural, meaning the feature availability is not exposed via CPUID and therefore not accessible by most OS commands like in this case 'resctrl' in Linux.

Intel configures a simple cache partitioning scheme using CAT when running real-time performance measurements. This simple scheme divides the cache in half, creating one half dedicated to the real-time application(s) and the other for best effort application(s). While not covered in this chapter beyond some basic allocation of cache to cores, additional details about using Cache Allocation Technology can be found in Chapter 5.

3.7 Resources for setting up Intel's connectivity supporting TSN

Setting up a TSN based network is outside the scope of this document. Intel offers some guidance for enabling TSN capabilities in a Linux environment using Intel's ethernet solutions in the Ethernet Time-Sensitive Networking on Linux* for Intel® Processors & Ethernet Controller i225/i226 – Get Started Guide which can be found in RDC, document number: 6164446 for customers under NDA.

3.8 Example: Configuring a 13th Generation Intel® Core™ i7-13800HRE platform for real-time

This section describes the high-level steps of preparing a 13th Generation Intel® Core™ platform for real-time usage.

Hardware selection:

- Intel® Core™ i7-13800HRE Processor
  - 14 Cores, 6 performance cores & 8 efficiency cores
  - 24MB of LLC (L3 Cache)
  - Integrated graphics
- Memory: Populate all DIMM slots
- Intel® Ethernet Controller I226 on PCIe port PEG (Port 1)

Configuring the Firmware

Enable TCC Mode in BIOS which will configure various BIOS menus for best real-time performance. In addition, it will configure all PCIe slots for Precision Time Measurement (PTM) and enable GT-CLOS, which limits the amount of cache the integrated graphics has access to.

Choosing and Configuring the Operating System

In the example, Real-time Ubuntu* optimized for Intel platforms is used.[1]

Apply kernel parameters to isolate cores for real-time workloads and correct clock usage. In this example we use P cores 4 & 5 for real-time usage and P-cores 0 - 3 as well as E-cores 6 - 13 for housekeeping and best effort workloads:

clocksource=tsc tsc=reliable nmi_watchdog=0 nosoftlockup idle=poll isolcpus=4-5 rcu_nocbs=4-5 nohz_full=4-5 irqaffinity=0[2]

Cache partitioning.

In this example we follow the simple cache partitioning scheme described previously, dividing the cache in half: one half dedicated to the real-time application(s) and the other for best effort application(s). The Intel® Core™ Processor i7-13800HRE Processor contains 24MB of LLC (L3) cache that can be partitioned using the model specific, non-architectural implementation of L3 Cache Allocation Technology

[1] Please contact Canonical for access to Real-time Ubuntu

[2] Follow Canonical's guidance on configuring other kernel boot parameter settings on real-time Ubuntu

found on Intel® Core™ processors that support Time Coordinated Computing. (See Figure 1)

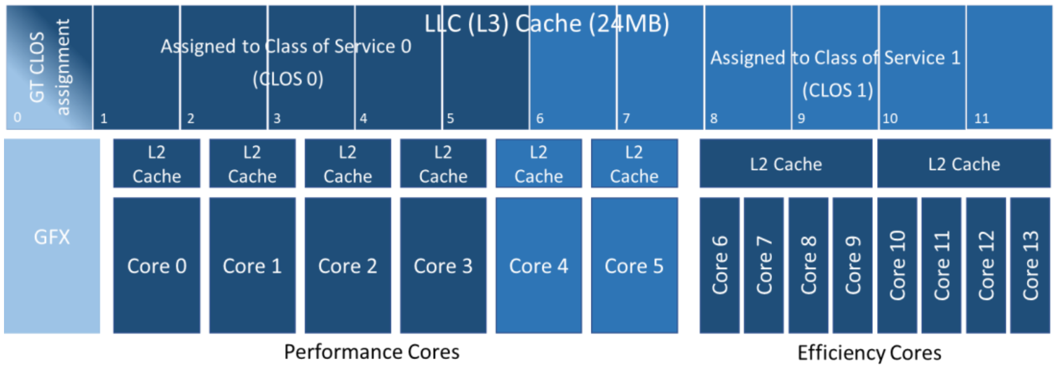

Figure 2 Cache allocation example for 13th Gen Intel® Core™ Processor i7-13800HRE example with GT CLOS enabled via TCC Mode in BIOS

Table 3 shows how the 24MB last level cache can be partitioned in half, with P cores 0-3, all E cores and Gfx operating with Class of Service 0 (CLOS 0) being assigned half the cache and P cores 4, and 5, which will be used for real-time workloads, operating on Class of Service 1 (CLOS 1) the other half. [1]

For completeness, the equivalent commands for programming the MSRs directly are also shown. The MSR's used for configuring the cache masks are L3_MASK_0 (address 0xC90) and L3_MASK_1 (address 0xC91). The MSR used for associating a class of service to a core is PQR_ASSOC (address 0xC8F).
The 13th Gen Intel® Core™ Processor i7-13800HRE last level cache has a capacity bitmask (CBM) length of 12 (0xFFF), so each class of service will be assigned a CBM length of 6 (0x3F and 0xFC0, respectively) when partitioning the cache in half.

The intel-cmt-cat "pqos" utility is an alternate way to program the cache allocation.

[1] Cache assignment to specific cores is done via Class of service (CLOS designation). Cache is assigned to a specific CLOS and cores run with a specific CLOS assigned to them.

Table 3: Partitioning the cache in half on the 13th Gen Intel® Core™ Processor i7-13800HRE.

Using intel-cmt-cat "pqos" utility	Model Specific Register (MSR) programming	Comments
pqos --iface=msr -e "llc:0=0x3F;llc:1=0xFC0"	wrmsr 0xC90 0x3F wrmsr 0xC91 0xFC0	Assign ½ the cache to CLOS 0 Assign ½ the cache (non-overlapping) to CLOS 1
pqos --iface=msr -a "llc:0=0-3, 6-13;llc:1=4,5"	wrmsr -p 0 0xC8F 0x000000000 wrmsr -p 1 0xC8F 0x000000000 wrmsr -p 2 0xC8F 0x000000000 wrmsr -p 3 0xC8F 0x000000000 ... wrmsr -p 4 0xC8F 0x100000000 wrmsr -p 3 0xC8F 0x100000000	Assign CLOS 0 to Cores 0-3 (performance cores) Also assign CLOS 0 to all efficiency cores (6-13) using the same command (not shown) Assign CLOS 1 to Cores 4 and 5

Note: The "–iface=msr" extension is necessary to make the pqos command work on non-architectural CAT.

Pinning real-time workloads to specific cores.

During run-time any real-time workloads should be pinned to the cores designated for high priority workloads (in this example Cores 4 and 5). This applies to the actual real-time workloads as well as any needed IRQ interrupt handling for such workloads. For best performance you pin the real-time workload to one Core (e.g. Core 4) and the real-time IRQ handling to the other (Core 5).

Note: Please refer to guidance from your OS vendor on how to pin workloads to cores.

3.9 Intel® Speed Shift technology for edge computing to boost real-time performance.

Setting the frequency of cores that have real-time workloads pinned to them to a level higher than base frequency (HFM) using Intel® Speed Shift technology can boost the real-time performance of those workloads. However, this benefit comes at a cost. Running cores above the HFM frequency impacts reliability and, to stay within the TDP guard bands, may cause more extensive throttling of other cores running best effort workloads.

On current Intel platforms the P-state of performance (P) cores can be selected independently per core[1]. Efficiency (E) cores are typically grouped in sets of four cores per module and the P-state can be selected per module.

When set correctly, the real-time cores will constantly run at the set frequency, while other cores will still vary their frequencies as directed by the processor or the OS, including the possibility of throttling as needed. It is important to note that the real-time cores can still be throttled if configured incorrectly. The figure below shows the behavior of various cores. In this example, two cores, labeled RT cores, are set to 3.6Ghz, the rest of the cores are 'floating' up to their set max-frequency values, in this example, p-cores up to 2.5GHz, E-cores up to 1.2Ghz, and the integrated GPU up to 1.3GHz.

[1] Future products may group p-cores into modules.

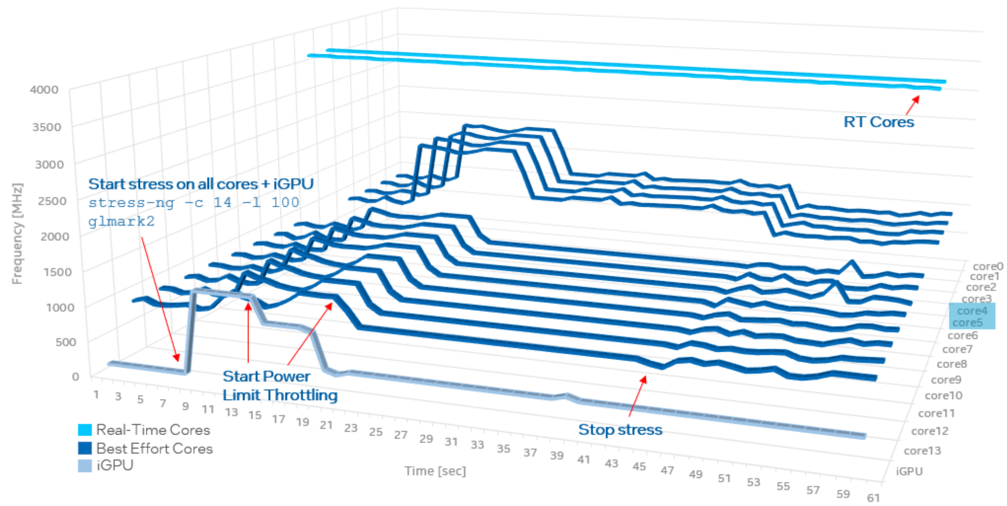

Figure 3: Core behavior when using Intel® Speed Shift technology for edge computing when system is loaded to induce throttling.

Setting some cores to such a higher frequency allows for lower average latencies and reduced jitter for the real-time workloads running on those cores.

3.9.1 Impact of using Intel® Speed Shift technology for edge computing on component reliability and overall system performance.

Setting even just a few cores to a higher, fixed frequency does not come without a cost. Due to higher internal frequency, voltages, and subsequent higher temperature and power, such settings will negatively impact the reliability expectations of the CPU and should be used with careful consideration.

Higher temperatures and power consumption also affects the overall system TDP and depending on the usage may adversely affect the performance of the rest of the cores and the graphics as the processor and OS will automatically throttle those cores and the graphics engine to avoid damage.

To address such concerns, we are describing a specific enveloping configuration on the highest performing SKU of a swim lane that can be leveraged into lower performing SKUs of the same processor family without significant impact on reliability. Such an enveloping configuration is described in the next section.

3.9.2 Example: Configuring Intel® Core™ i7-13800HRE for the enveloping Intel® Speed Shift for edge compute use condition.

To use Intel® Speed Shift Technology for edge computing, it is essential to enable it in the BIOS. The following BIOS settings are recommended:

Table 4: BIOS seeting recommendations.

Ref BIOS menu	BIOS setting	State	Comment
Intel Advanced Menu > Power & Performance > CPU – Power Management Control	Turbo Boost	Enabled	When disabled, turns off Intel® Turbo Boost Technology.
	Intel Speed Shift Technology	Enabled	When disabled, turns off Intel® Speed Shift Technology.
	Intel Speed Step	Enabled	When disabled, turns off Intel Speed Step® technology.
	Energy Efficient Turbo	Disabled	When enabled, turbo frequency is opportunistically lowered to increase efficiency
	HWP Autonomous EPP Grouping	Disabled	When enabled, HWP autonomous requests same p-state for all cores with same EPP value
Intel Advanced Menu > Power &Performance > GT Power Management Control	GT Max Turbo Frequency	300MHz	When set, the GT Max Turbo Frequency is set to the selected value.

Software can detect Intel® Speed Shift technology support using the CPUID instruction. Below are the model-specific registers (MSRs) to be used to configure P-Cores and E-cores with Intel® Speed Shift technology for this specific example. For additional information on MSRs used reference the Intel® 64 and IA-32 Architectures Software Developer Manuals.

MSR details	Setting	Comment
HWP Enable MSR Register: IA32_PM_Enable Address: 0x770	0x1	Switches on Hardware P-States. Sticky bit which can be written once and will stay to the next RESET
HWP Capabilities MSR Register: IA32_HWP_CAPABILITIES Address: 0x771		Lists HWP performance range
HWP Request MSR Register: IA32_HWP_REQUEST Address: 0x774	#P-Cores core 0: 0xf000000080002006 core 1: 0xf000000081002006 core 2: 0xf000000082002006 core 3: 0xf000000083002006 core 4: 0xf0000000002e2e2e core 5: 0xf0000000002e2e2e #E-Cores core 6: 0xf000000085001204 core 7: 0xf000000086001204 core 8: 0xf000000087001204 core 9: 0xf000000088001204 core 10: 0xf000000089001204 core 11: 0xf00000008a001204 core 12: 0xf00000008b001204 core 13: 0xf00000008c001204	Sets the frequencies and preferences per core. Used by the OS to provide hints (min, max, desired, energy performance preferences to HWP)

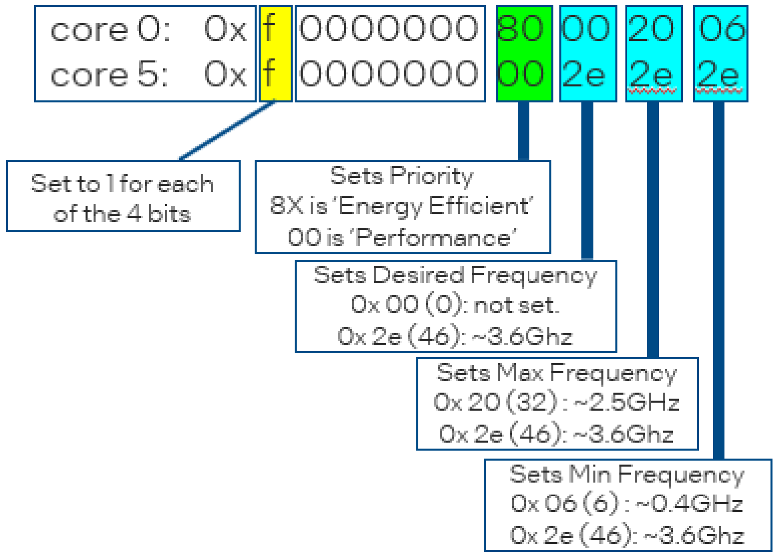

Figure 4:Example: Intel® Speed Shift technology setting per core frequencies.

Generally, the following formulas should be used to arrive at the right frequency settings:

Performance Cores:

'Desired Frequency in GHz' * 10 *1.27			**= Decimal-**	**→ Hex- Setting**
2.5 GHz example:	2.5	* 10 *1.27	= 32	→ 0x20
3.6 GHz example:	3.6	* 10 *1.27	= 46	→ 0x2e
0.4 GHz example:	0.4	* 10 *1.27	= 6	→ 0x06

Efficiency Cores:

'Desired Frequency in GHz' * 10 *1.0			**= Decimal-**	**→ Hex-Setting**
1.8 GHz example:	1.8	* 10 *1.0	= 18	→ 0x12
0.4 GHz example:	0.4	* 10 *1.0	= 4	→ 0x04

To confirm the core is running at the desired frequency a publicly available tool such as Turbostat on Linux, or corresponding tools on other OS. Likewise, you may calculate the average effective frequency $\overline{f_{ij}}$ within a time span $\Delta_{ij}t = t_j - t_i$ between two consecutive points in time t_i, and t_j for a certain core via

$$\overline{f_{ij}} = \frac{\Delta_{ij}A}{\Delta_{ij}M} f_{tsc}$$

where $\Delta_{ij}A$, and $\Delta_{ij}M$ are the differences between the values APERF, and MPERF at t_i, and t_j each, and f_{tsc} the tsc frequency reported by the OS.

For more information on APERF, MPERF, and TSC Frequency please refer to the Intel SDM vol 3B, Chapter "HARDWARE DUTY CYCLING (HDC) - MPERF and APERF Counters Under HDC".

The amount of throttling occurring on other cores heavily depends on the total load applied by the customer as well as the thermal solution used. If needed, customers should evaluate using lower frequency settings than the ones described in this enveloping configuration. Especially for the graphics it may be best to set the frequency to the lowest value that fulfills the workload requirements to reduce throttling of the LLC/Ring.

As mentioned in the previous section, using Intel® Speed Shift technology in that manner may impact reliability and overall system performance. For customers under NDA, Intel provides a reliability assessment for the described configuration. This information is available in the Intel® TCC User Guide (RDC #786715). This allows customers to confidently configure their systems to increase the performance of real-time workloads running on specific cores when setting the Max-frequencies at or below the prescribed values, 3.6GHz for up to two cores and HVM on all remaining cores and GFX. While Intel® Turbo Technology is enabled, cores can only reach the prescribed max frequencies, never reaching the much higher turbo frequency of that processor. Depending on the thermal solution used and the workloads running, lower Max-frequencies may be more suitable to avoid throttling of the non-RT cores.

3.10 Timed General Purpose I/O (TGPIO)

Sometimes it's not enough for software to be told what the current OS/CPU time is "now". For example, when precision time synchronization is a regulatory requirement, it can be necessary to verify (at least in the lab) the accuracy of the CPU's notion of "now" using a hardware signal as a ground truth reference. Intel has addressed this need by incorporating Timed General-Purpose I/O (TGPIO) pins on the CPU subsystem that generate a rising digital logic signal precisely when the CPU time reaches a specified time.

TGPIO pins are linked to the host CPU's hardware clock, enabling both the import/export of the unified time domain to/beyond the host as well as validation of time at the host. TGPIO pins can be used to import a Pulse Per Second (PPS) signal to discipline the system clock or generate a PPS output to synchronize other devices that lack PTP capabilities. Additionally, TGPIO serves as a critical tool for verifying system-level timing accuracy when used in conjunction with a signal analyzer. By comparing the PPS output from a TGPIO pin to a similar PPS output from a reference time source, engineers can assess the accuracy of the CPU time.

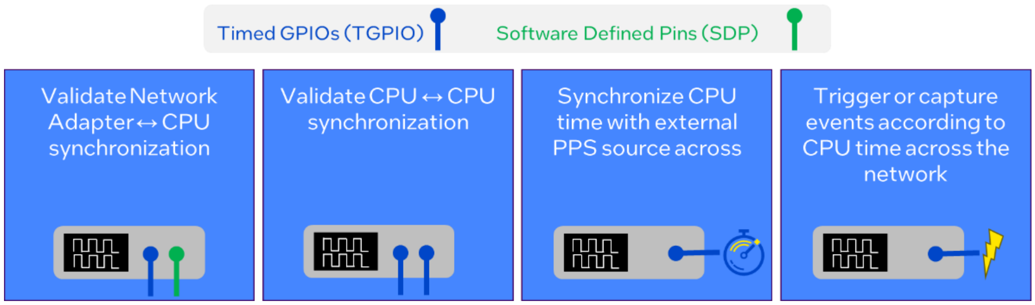

Figure 5: TGPIO usage scenarios.

Here are several practical examples of TGPIO usage:

1. By connecting a TGPIO pin and the Software-Defined Pin (SDP) of an Intel® Ethernet Controller i226 to a logic analyzer, test engineers can measure the time difference between the two signals. When the CPU and PCH clocks are synchronized, the PPS logic transitions will occur nearly simultaneously. This setup is particularly valuable for evaluating the time accuracy difference between a network adapter and the CPU, especially when comparing scenarios with PCIe PTM enabled versus disabled.
2. By comparing TGPIO signals from two end systems, engineers can gain insights into the end-to-end system error, enhancing their understanding of overall system synchronization.
3. Synchronize time between the CPU and another hardware component, such as a GPS module using TGPIO as an input
4. TGPIO PPS output can be used to control sensor sampling to ensure sensor data acquisition of different sensors happen simultaneously.

The TGPIO pin can be found in the platform GPIO table with the native function TIME_SYNC0 or TIME_SYNC1.

Note 1: When enabling TGPIO PPS function, the crystal oscillator will not be shut down as crystal clock is needed for the Timed GPIs operation. As a result, SLP_S0# will not be asserted for low power mode. Software should only enable Timed GPIO when needed and disable it when Timed GPIO functionality is not required.

Note 2: Intel® Ethernet Network Adapters such as I226 also can support PPS input/output with SDP pins. Please notice the default pin voltage level and polarity are different between PCH TGPIO to NIC SDP. Currently TGPIO drivers are only available on Linux. Upstream TGPIO driver for PPS output

Intel's TGPIO driver triggers a pulse per second (PPS) continuously when installed, allowing engineers to check synchronization or align synchronization to the signal on the TGPIO pin. The TGPIO PPS output driver is available in mainline Linux (https://www.kernelconfig.io/config_pps_generator_tio).

The following example instructions describe how to verify the presence of the PPS driver and enable or disable PPS signal generation.

Verifying the PPS driver is present:n$ modinfo pps_gen_tio | grep description
 # description: Intel PMC Time-Aware IO Generator Driver

Enabling/Disabling PPS Generation:
If the SYSFS filesystem is enabled in the kernel it provides a new class:
$ ls /sys/class/pps-gen/
 # pps-gen0

Every directory is the ID of a PPS generator defined in the system and continue various files and control structures:
$ ls -F /sys/class/pps-gen/pps-gen0/
 # dev enable power/ subsystem@ system time uevent

To enable the PPS signal generation:
$ echo 1 > /sys/class/pps-gen/pps-gen0/enable

To disable the PPS signal generation:
$ echo 0 > /sys/class/pps-gen/pps-gen0/enable

3.10.1 Out-of-tree TGPIO driver

There is an OOT driver in the Intel Ubuntu Overlay BKC for supporting both TGPIO output and input:

- drivers/ptp/ptp-intel-pmc-tgpio.c for all platforms except Intel Atom® x6000E
- drivers/ptp/ptp-intel-tgpio-plat.c for Intel Atom® x6000E only.
- For further information on how to use this driver, please see RDC#781241 TGPIO_Validation_Methodology.

4.0 Terminology

Terminology

Term	Description
API	Application Programming Interface
ASPM	Active State Power Management
BKC	Best Known Configuration
BSP	Board Support Package
C-State	A core power state requested by the Operating System Directed Power Management (OSPM) infrastructure that defines the degree to which the process is “sleeping”.
CAT	Cache Allocation Technology
CBM	Capacity Bit Mask
CMF	Coherent Memory Fabric
COS	Class of Service
CPS	Cyber-physical systems
CRBs	Customer Reference Boards
Deadline	The time when some computation or data must complete or arrive. For some applications, computations or data that arrive late are no longer useful.
E2E	End-to-end
ECC	Error-correcting-code
EDS	External Design Specification
FuSA	Functional Safety
IFWI	Integrated Firmware Image
IHS	Integrated Heat Spreader
Intel® RDT	Intel® Resource Director Technology
TCC	Time Coordinated Computing A modern approach to architecting distributed, synchronized, scalable computing systems that address real-time application requirements for cyber-physical systems (CPS). TCC moves beyond traditional real-time systems based on simple microcontrollers.

Term	Description
IOSF	Input/Output Scalable Fabric
IoT	Internet of Things
Jitter	The difference between the maximum and minimum of some quantity, such as latency measured in units of time. Jitter matters a lot at sensors and actuators, but other mechanisms (such as TSN mechanisms) typically hide software-execution jitter (so long as WCET bounds are satisfied).
KPI	Key Performance Indicator
LCC	Low Core Count
L2	Level 2 cache
L3/LLC	Last level cache
Latency	The duration of time between two events; for example, the time a signal is detected, and a response is received, or the time between an application sending a UDP message until it arrives on the Ethernet wire, or the time required to execute a code segment.
MSR	Model-specific registers are a group of registers available primarily for the operating-system or executive procedures (that is, code running at privilege level 0). These registers control items such as the following: • debug extensions • performance-monitoring counters • machine- check architecture • memory type ranges (MTRRs)
MMIO	Memory Map IO
MVC	Multi Virtual Channel
Noisy neighbor	An application or device, the functioning of which, affects the device or application with temporal requirements (e.g., because of shared resources). Temporal Isolation seeks to reduce the deleterious effect of a noisy neighbor.
OPC UA	A platform-agnostic standard for communication between devices using an "Unified Architecture", created by the OPC Foundation focusing on the needs of industrial automation.
OSPM	Operating System Directed Power Management
P-State	A power-performance, implantation-dependent state of devices or processors that indicate power and frequency levels.

Term	Description
PCH	Platform Controller Hub
PCS	Physical Coding Sublayer
PCIe*	Peripheral Component Interconnect express*
PM	Power Management
PPS	Pulse Per Second
PTM	Precision Time Measurement, specified by the PCI SIG
PREEMPT_RT	Before version 6.12, in order to improve the real-time performance of Linux, different changes had to be applied to the source code, grouped under the so called "PREEMPT_RT patch", named after the PREEMPT_RT preemption model that it introduced. From 6.12 onwards, the real-time optimizations are fully integrated and the PREEMPT_RT preemption model can be selected in the official upstream kernels. After 6.12, the PREEMPT_RT patch is still maintained for previous versions, and used as a staging area for changes not yet upstreamed.
RCU	Read-Copy-Update
RDC	Resource & Design Center
Real-time application	An application that requires a complete execution within some WCET with a specified level of reliability. For example, "Has to finish running every millisecond without missing a deadline in 7 days." Typically, a real-time application contains a sequence of three tasks: sense > compute > actuate and increasingly uses a network to interconnect these. The extent to which a missed deadline impacts the overall system is sometimes described using "soft", "firm", and "hard" real-time, but we avoid these terms, preferring to quantify the reliability with number of 9s.
RTCP	Real-Time Compute Performance (Intel developed real-time performance benchmark)
RTOS	Real-Time Operating System
SA	System Agent
SGMII	Serial Gigabit Media-Independent Interface
SKU	Stock Keeping Unit
STA	Station Management
TC	Traffic Classes

Term	Description
TDP	Thermal Design Power
Temporal Isolation	The degree to which a system can meet the time-related requirements of a real-time workload when the workload is running alongside other workloads on the system. In a typical system, concurrent workloads create contention for shared resources that can cause spikes in latency and increased jitter for the real-time workload. TCC capabilities help mitigate concurrent workload interference.
Time-Sensitive Networking (TSN)	A task-group of IEEE 802.1 that creates / amends the Ethernet and other standards, enabling dramatically better worst-case time performance (time-synchronization & latency). Also used to describe the standards / amendments from the TSN task group. https://1.ieee802.org/tsn/
Time synchronization	The degree to which two or more systems or devices agree on what time it is, to within some maximum error. Enables sensors, compute systems, actuators, and network elements to operate on a global schedule. Enables time-coordinated computing devices to time-division multiplex real-time and non-real-time tasks, measure latencies, and detect violation of deadlines. Enables an RTOS to schedule a task at a specific time.
TGPIO	Timed-General Purpose I/O
UDP	User Datagram Protocol
UEFI	Unified Extensible Firmware Interface
VCs	Virtual Channels
Workload	An application that performs some useful computational work, including (perhaps) receiving input, performing computation, and generating an output.
WCET	Worst-case execution time. The maximum measured latency of the compute portion of an application, across multiple iterations. WCET relates to reliability, described by “the number of 9s”. For instance, reliability of two 9s refers to missing the deadline 1 out of 100 times.

§

5.0 Reference Documents

In Table 5, documents with a document number are available on Resource and Design Center. Log into the Resource and Design Center to search for and download the document numbers. Contact your Intel field representative for access.

Note: Third-party links are provided as a reference only. Intel does not control or audit third-party benchmark data or the websites referenced in this document. Visit the referenced website and confirm whether the referenced data is accurate.

Table 5: Reference Documents

Document	Document No./Location
NDA Version of the Intel® TCC User Guide	786715 https://cdrdv2.intel.com/v1/dl/getContent/786715
Real-Time Gold Deck	627170 https://cdrdv2.intel.com/v1/dl/getContent/627170
Real-Time Key Performance Indicators (KPIs)	738816 https://cdrdv2.intel.com/v1/dl/getContent/738816
Ethernet Time-Sensitive Networking on Linux* for Intel® Processors & Ethernet Controller i225/i226 – Get Started Guide	616446 https://cdrdv2.intel.com/v1/dl/getContent/616446
Wiki for Intel-cmt-cat on Github	intel/intel-cmt-cat Wiki (github.com)
13th Gen Intel® Core™ Mobile Processors for IoT Edge Page	https://www.intel.com/content/www/us/en/products/details/embedded-processors/core/13thgenmobile.html
14th Gen Intel® Core™ Desktop Processors for IoT Edge Page	https://www.intel.com/content/www/us/en/products/details/embedded-processors/core/14thgen.html
Intel® Ethernet Controller I226 Collection	https://www.intel.com/content/www/us/en/products/details/ethernet/gigabit-controllers/i226-controllers/docs.html?wapkw=i226&grouping=rdc+Content+Types&s=Newest

Document	Document No./Location
Intel® Ethernet Controller I225/I226 Product Brief	621753 https://cdrdv2.intel.com/v1/dl/getContent/621753
Intel® 64 and IA-32 Architectures Software Developer's Manual	https://software.intel.com/en-us/articles/intel-sdm
Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 3B: System programming guide, part 2	671427 https://cdrdv2.intel.com/v1/dl/getContent/671427
Get Started with Linux TSN	https://docs.google.com/document/d/1va_VbXEuVglQR2Xvawd01buUjhs5wifDw0tZ8xI56yE/edit?usp=sharing This User Guide describes the Time-Sensitive Networking (TSN) Reference Software for Linux*. It includes three demos and walks users through running them as well as understanding the features and capabilities of this reference software.
Yocto Project* Linux Kernel Development Manual	https://www.yoctoproject.org/docs/1.6.1/kernel-dev/kernel-dev.html
PCI-SIG, PCI Express Base Specification Revision 2.1 (subscription based)	http://pcisig.com/specifications/pciexpress/base
Linux kernel command-line parameters	https://www.kernel.org/doc/html/latest/admin-guide/kernel-parameters.html
Event Histograms	https://git.kernel.org/pub/scm/linux/kernel/git/rt/linux-stable-rt.git/tree/Documentation/trace/histogram.rst?h=v5.15-rt
The Linux Foundation’s Real Time Linux (PREEMPT_RT) collaborative project	https://wiki.linuxfoundation.org/realtime/start
Technical Paper: Empowering Mixed-criticality Industrial Real-time Computing on Performance Hybrid Architecture with Intel's Dynamic Frequency Scaling Evolution	816106 Empowering Mixed-criticality Industrial Real-time Computing on Performance Hybrid Architecture with Intel's Dynamic Frequency Scaling Evolution

Appendix A - Intel processors with Time Coordinated Computing Technology support

Note: Starting in 2025, real-time capabilities (including TCC and TSN) can be enabled across all Edge recommended products. This appendix provides information on TCC support on earlier platforms. For the latest information on Edge Computing roadmap and capabilities please refer to the Intel Edge Compass RDC # 350329.

A.1 Intel® Xeon® Processors with Intel® TCC support

Note: For the latest details on specific SKUs and platform capabilities, please refer to the information provided at: Intel® Xeon® Processors

A.1.1 Intel® Xeon® D-2700 and D-1700 Processors

The following SKUs provide support for Time Coordinated Computing Technology (check intel.com for the most up-to-date product information).

Intel® Xeon® D-2700 and D-1700 processors with Intel® TCC support.

	D-2752TER	D-1746TER	D-1735TR	D-1715TER	D-1712TR
Use Condition	Embedded	Embedded	Embedded	Embedded	Embedded
Cores	12	10	8	4	4
TDP	77W	67W	59W	50W	40W
TCC	Yes	Yes	Yes	Yes	Yes

A.1.2 Intel® Xeon® W-11000E Series Processors

The following SKUs provide support for Intel® TCC (check intel.com for the most up-to-date product information).

Intel® Xeon® W-11000E Series processors with Intel® TCC support

	W-11865MRE	W11865MLE
Use Condition	Industrial	Industrial
Cores	8	8

	W-11865MRE	W11865MLE
TDP	45W	25W
TCC	Yes	Yes

	W-11555MRE	W-11555MLE	W-11155MRE	W-11155MLE
Use Condition	Industrial	Industrial	Industrial	Industrial
Cores	6	4	4	6
TDP	45W	25W	35W	25W
TCC	Yes	Yes	Yes	Yes

A.2 Intel® Core™ Processors with Intel® TCC support

Note: For the latest details on specific SKUs and platform capabilities, please refer to the information provided at: Intel® Core™ Processors

A.2.1 14th Generation Intel® Core™ Processors (S Series)

The following SKUs provide support for Intel® TCC (check intel.com for the most up-to-date product information).

14th Generation Intel® Core™ S-series Processors with Intel® TCC support (Corporate/Mainstream)

	i9-14900/ 14900T	i7-14700/ 14700T	i5-14500/ 14500T	i5-14400/ 14400T	i3-14100/ 14100T
Use Condition	PC Client	PC Client	PC Client	PC Client	PC Client
Cores[1]	24 (8 + 16)	20 (8 + 12)	14 (6 + 8)	10 (6 + 4)	8 (4 + 0)
TCC	Yes	Yes	Yes	Yes	Yes

	i9-14901E	I7- 14701E	I5- 14501E	I5- 14401E
Use Condition	PC Client	PC Client	PC Client	PC Client
Cores[2]	8 P-cores	8 P-cores	6 P-cores	6 P-cores
TCC	Yes	Yes	Yes	Yes

14th Generation Intel® Core™ S-series Processors with Intel TCC support (Low Power)

	I9-13900T	i7-14700T	i5-14500T	i5-14400T	i3-14100T
Use Condition	PC Client	PC Client	PC Client	PC Client	PC Client
Cores[1]	24 (8 + 16)	20 (8 + 12)	14 (6 + 8)	10 (6 + 4)	8 (4 + 0)
TCC	Yes	Yes	Yes	Yes	Yes

[1] Processor cores listed first are the total number of cores in the processor followed by the number of Performance-cores and number of Efficient-cores in parentheses.
[2] Processor cores listed first are the total number of cores in the processor followed by the number of Performance-cores and number of Efficient-cores in parentheses.

	i9-14901TE	I7-14701TE	I5-14501TE	I514401TE
Use Condition	PC Client	PC Client	PC Client	PC Client
Cores[1]	8 P-cores	8 P-cores	6 P-cores	6 P-cores
TCC	Yes	Yes	Yes	Yes

A.2.2 13th Generation Intel® Core™ Processors (U/P/H Series)

The following SKUs provide support for Intel® TCC (check intel.com for the most up-to-date product information).

13th Generation Intel® Core™ U-Series Processors (15W) with Intel® TCC support

	i7-1365UE	i5-1345UE	I5-1335UE	i3-1315UE
Use Condition	Gen Embedded	Gen Embedded	Gen Embedded	Gen Embedded
Cores[1]	10 (2 + 8)	10 (2 + 8)	10 (2 + 8)	6 (2 + 4)
TCC	Yes	Yes	Yes	Yes

	i7-1365URE	i5-1345URE	i3-1315URE	I7-1366URE (Fusa)
Use Condition	Industrial	Industrial	Industrial	Industrial
Cores[1]	10 (2 + 8)	10 (2 + 8)	6 (2 + 4)	10 (2 + 8)
TCC	Yes	Yes	Yes	Yes

13th Generation Intel® Core™ P-Series Processors (28W) with Intel® TCC support

	i7-1370PE	i5-1350PE	i5-1340PE	i3-1320PE
Use Condition	Gen Embedded	Gen Embedded	Gen Embedded	Gen Embedded
Cores[1]	14 (6 + 8)	12 (4 + 8)	12 (4 + 8)	8 (4 + 4)
TCC	Yes	Yes	Yes	Yes

[1] Processor cores listed first are the total number of cores in the processor followed by the number of Performance-cores and number of Efficient-cores in parentheses.

	i7-1370PRE	i5-1350PRE	i3-1320PRE	I7-1375PRE (Fusa)
Use Condition	Industrial	Industrial	Industrial	Industrial
Cores[1]	14 (6 + 8)	12 (4 + 8)	8 (4 + 4)	14 (6 + 8)
TCC	Yes	Yes	Yes	Yes

13th Generation Intel® Core™ H-Series Processors (45W) with Intel® TCC support

	i7-13800HE	i5-13600HE	i3-13300HE
Use Condition	Gen Embedded	Gen Embedded	Gen Embedded
Cores[1]	14 (6 + 8)	12 (4 + 8)	8 (4 + 4)
TCC	Yes	Yes	Yes

	i7-13800HRE	i5-13600HRE	i3-13300HRE
Use Condition	Industrial	Industrial	Industrial
Cores[1]	14 (6 + 8)	12 (4 + 8)	8 (4 + 4)
TCC	Yes	Yes	Yes

A.2.3 13th Generation Intel® Core™ Processors (S Series)

The following SKUs provide support for Intel® TCC (check intel.com for the most up-to-date product information).

13th Generation Intel® Core™ S-series Processors with Intel® TCC support (Corporate/Mainstream)

	i9-13900	i7-13700	i5-13500	i5-13400	i3-13100
Use Condition	PC Client	PC Client	PC Client	PC Client	PC Client
Cores[1]	24 (8 + 16)	16 (8 + 8)	14 (6 + 8)	10 (6 + 4)	4 (4 + 0)
TCC	yes	Yes	Yes	Yes	Yes

[1] Processor cores listed first are the total number of cores in the processor followed by the number of Performance-cores and number of Efficient-cores in parentheses.

	i9-13900E	i7-13700E	i5-13500E	i5-13400E	i3-13100E
Use Condition	Embedded	Embedded	Embedded	Embedded	Embedded
Cores[1]	24 (8 + 16)	16 (8 + 8)	14 (6 + 8)	10 (6 + 4)	4 (4 + 0)
TCC	Yes	Yes	Yes	Yes	Yes

13th Generation Intel® Core™ S-series Processors with Intel® TCC support (Low Power)

	i7-13700T	i5-13500T	i3-13100T
Use Condition	PC Client	PC Client	PC Client
Cores[1]	16 (8 + 8)	14 (6 + 8)	4 (4 + 0)
TCC	Yes	Yes	Yes

	i9-13900TE	i7-13700TE	i5-13500TE	i3-13100TE
Use Condition	Embedded	Embedded	Embedded	Embedded
Cores[1]	24 (8 + 16)	16 (8 + 8)	14 (6 + 8)	4 (4 + 0)
TCC		Yes	Yes	Yes

A.2.4 12th Generation Intel® Core™ Processors (S-Series)

The following SKUs provide support for Intel® TCC (check intel.com for the most up-to-date product information).

12th Generation Intel® Core™ S-series Processors with Intel® TCC support

	i9-12900E	i7-12700E	i5-12500E	i3-12100E
Use Condition	Embedded	Embedded	Embedded	Embedded
Cores[2]	16 (8 + 8)	12 (8 + 4)	6 (6 + 0)	4 (4 + 0)
TDP	65W	65W	65W	65W
TCC	Yes	Yes	Yes	Yes

[1] Processor cores listed first are the total number of cores in the processor followed by the number of Performance-cores and number of Efficient-cores in parentheses.
[2] Processor cores listed first are the total number of cores in the processor followed by the number of Performance-cores and number of Efficient-cores in parentheses.

A.2.5 11th Generation Intel® Core™ Processors (UP3 Series)

The following SKUs provide support for Intel® TCC (check intel.com for the most up-to-date product information).

11th Generation Intel® Core™ UP3-Series Processors with Intel® TCC support

	i7-1185GRE	I5-1145GRE	I3-1115GRE
Use Condition	Industrial	Industrial	Industrial
Cores	4	4	2
TDP	28W	28W	28W
TCC	Yes	Yes	Yes

A.3 Intel Atom® Processors with Intel® TCC support

Note: For the latest details on specific SKUs and platform capabilities, please refer to the information provided at: [Intel Atom® Series Processors]

A.3.1 Intel Atom® x7000E, x7000RE, and x7000C Series Processors

The following SKUs provide support for Intel® TCC (check intel.com for the most up-to-date product information).

Intel Atom® x7000RE Series Processors with Intel® TCC support (Industrial)

	X7211RE	X7213RE	X7433RE	X7835RE
Use Condition	Embedded, Industrial, and Communication			
Cores	2	2	4	8
TDP	6W	9W	9W	12W
TCC	Yes	Yes	Yes	Yes
FuSa	No	No	No	No

Intel Atom® x7000C Series Processors with Intel® TCC support (Communication)

	X7203C	X7405C	X7809C
Use Condition	Embedded and Communication		
Cores	2	4	8
TDP	9W	12W	25W
TCC	Yes	Yes	Yes
FuSa	No	No	No

Intel Atom® x7000E Series Processors with Intel® TCC support

	X7211E	X7425E	X7213E
Use Condition	Embedded		
Cores	2	4	2
TDP	6W	12W	10W
TCC	Yes	Yes	Yes

	X7211E	X7425E	X7213E
FuSa	No	No	No

A.3.2 Intel Atom® x6000E Series Processors

The following SKUs provide support for Intel® TCC (check intel.com for the most up-to-date product information).

Intel Atom® x6000E Series Processors with Intel® TCC support

	x6212RE	X6414RE	X6425RE	X6200FE	X6427FE
Use Condition	Industrial	Industrial	Industrial	Industrial	Industrial
Cores	2	4	4	2	4
TDP	6W	9W	12W	4.5W	12W
TCC	Yes	Yes	Yes	Yes	Yes
FuSa	No	No	No	Yes	Yes

Intel Atom® x6000E Series Performance Upgrade Processors with Intel® TCC support

	X6214RE	X6416RE
Use Condition	Industrial	Industrial
Cores	2	4
TDP	6W	9W
TCC	Yes	Yes
FuSa	No	No

§