13th Generation Intel® Core™ Processors

Real-Time Tuning Guide

February 2023

Intel Confidential

Document Number: 757534-0.7

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit www.intel.com/design/literature.htm.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No product or component can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

The code names presented in this document are only for use by Intel to identify products, technologies, or services in development, that have not been made commercially available to the public, i.e., announced, launched or shipped. They are not "commercial" names for products or services and are not intended to function as trademarks.

Contents

1.0 **Introduction** 7

1.1 About This Document 8

2.0 **Capabilities Overview** 9

2.1 Hardware Features 9

2.2 Supported Hardware 9

2.2.1 13th Generation Intel® Core™ Processors 9

2.2.2 Where to Get Hardware 12

2.3 Intel® TCC Software 12

2.3.1 Application and Middleware Support 13

2.3.2 Operating System Support 13

2.3.3 Driver Support 13

2.3.4 BIOS Support 14

2.3.5 Slim Bootloader Support 15

2.4 Other Analysis and Profiling Tools 16

2.5 Time-Sensitive Networking Software 16

2.6 Best Known Configuration (BKC) 17

3.0 **Real-Time Key Performance Indicators (KPIs)** 19

4.0 **Intel® TCC Platform Tuning Strategy** 20

5.0 **System Software Tuning** 22

5.1 Linux Operating Systems 22

5.1.1 Kernel Parameters 22

5.1.2 Other Linux OS Optimizations 26

5.1.3 Linux OS Resource Partitioning 27

6.0 **Platform Power Management Tuning** 30

6.1 Intel® TCC Mode (BIOS): Power Management Settings 30

6.2 System PM 33

6.2.1 Intel® Turbo Boost Technology (P0 States) 33

6.2.2 Enhanced Intel SpeedStep® Technology and Intel SpeedStep® (P-States) 34

6.2.3 C-States 34

6.2.4 S-States 34

6.2.5 Display C-States 34

6.3 I/O PM 35

6.3.1 I/O Fabric PM 35

6.3.2 Individual I/O System PM 35

7.0 **Intel® TCC Features Tuning** 36

7.1 Intel® TCC Mode (BIOS): Intel® TCC Feature Settings 36

7.1.1 Overview 36

7.1.2 Architecture 36

7.1.3 Usage....37
7.1.4 Intel® TCC Mode (BIOS): Intel® TCC Feature Settings....37
7.2 Cache Allocation Technology (CAT)....38
7.2.1 Overview....38
7.2.2 Architecture....38
7.2.3 Usage....38
7.3 Data Direct Input/Output (DDIO) / Write Cache (WRC)....40
7.3.1 Overview....40
7.3.2 Architecture....41
7.3.3 Usage....41
7.4 Upstream Virtual Channels....41
7.4.1 Overview....41
7.4.2 Architecture....42
7.4.3 Usage....45
7.5 Time Synchronization....45
8.0 Fabric Tuning....46
9.0 Terminology....47
10.0 Reference Documents....51
Appendix A Variant Kernel Configurations....54

Figures

Figure 1. Intel® TCC Reference Software Stack....12
Figure 2. Example of Intel® TCC Mode Interface....14
Figure 3. TSN Software....17
Figure 4. Fabric Virtual Channels Block Diagram....44

Tables

Table 1. 13th Generation Intel® Core™ S-Series Processors with Intel® R680E Platform Controller Hub (PCH) – Corporate/Mainstream....10
Table 2. 13th Generation Intel® Core™ S-Series Processors with Intel® R680E Platform Controller Hub (PCH) – Low Power....10
Table 3. 13th Generation Intel® Core™ U-Series Processors (15W)....10
Table 4. 13th Generation Intel® Core™ P-Series Processors (28W)....10
Table 5. 13th Generation Intel® Core™ H-Series Processors (45W)....10
Table 6. Other Analysis and Profiling Tools....16
Table 7. Other Analysis and Profiling Tools....22
Table 8. Kernel Command-Line Parameters....23
Table 9. Intel® TCC Mode (BIOS): Power Management Settings....30

Table 10. Intel® TCC Mode (BIOS): Intel® TCC Feature Settings 37
Table 11. L3 CAT CBMs 39
Table 12. PCIe Controller Virtual Channel Support 42
Table 13. PCIe Virtual Channel Mapping Capabilities 43
Table 14. Terminology 47
Table 15. Reference Documents 51
Table 16. Kernel Command-Line Parameters for RTCP/XDP and Cyclic Test 54

Revision History

Date	Revision	Description
February 2023	0.7	Initial release.

§

1.0 Introduction

Intel processors are multi-purpose and can serve a wide range of use cases from data analysis in the cloud, to gaming PCs and traditional office laptops, to devices at the edge.

The 13th Generation Intel® Core™ Processors are new members of the Intel® Time Coordinated Computing (Intel® TCC) family.

Intel® TCC provides the following key values:

- A set of features that augment the compute performance of its processors with ability to address the stringent temporal requirements of real-time applications.
- Improved temporal performance for latency-sensitive applications when they are running alongside non-time-constrained applications on the same system.

Note: The current version of this document only describes real-time features and optimizations of the 13th Generation Intel® Core™ Processors. Some features are not available on all product stock keeping units (SKUs). Refer to 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.
3. Control an actuator or render a result.

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

For the 13th Generation Intel® Core™ Processors, Intel provides the following support for real-time applications:

- Selected processors with features to minimize worst-case execution time (WCET) within the system, referred to as Intel® TCC features.
- Discrete Ethernet controllers that support IEEE 802.1 Time-Sensitive Networking (TSN) standards.
- Best known configurations (BKC) with real-time optimizations. The BKC is the foundation of Intel's key performance indicators (KPIs) of time performance.

- Software tools to accelerate hardware configuration/tuning and application development by enabling performance analysis and access to hardware features.

1.1 About This Document

This document describes tuning for real-time applications. Tuning covers every part of the system, including network, hardware, BIOS, operating system, and the application itself. This document is for anyone working on these components, such as system engineers, firmware developers, operating system developers, and application developers.

Consider this document the starting point for understanding real-time tuning. This document:

- Introduces real-time hardware features and related software.
- Introduces the KPIs that Intel uses to measure real-time performance.
- Describes the tuning strategy, including details about use of real-time features and analysis/profiling tools.

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 Capabilities Overview

2.1 Hardware Features

The IoT technology segment increasingly requires time-related capabilities both within embedded devices and across networks to support smart or automated cyber-physical systems (CPS).

Supported 13th Generation Intel® Core™ Processors provide Intel® Time Coordinated Computing (Intel® TCC) features to optimize real-time compute performance. Supported Ethernet controllers provide IEEE TSN features to optimize network traffic.

In this document, the hardware features are described in relation to real-time tuning. These features include Cache Allocation Technology (CAT), upstream virtual channels, and more. For information about hardware features beyond the scope of this document, see the external design specification *EDS Addendum* listed in the Reference Documents.

2.2 Supported Hardware

2.2.1 13th Generation Intel® Core™ Processors

This document covers Intel® Core™ S-Series as well as Intel® Core™ U/P/H-Series processors. Intel recommends the following processors for real-time applications or solutions. Intel® Core™ S-Series processors need to be paired with Intel® R680E Platform Controller Hub (PCH):

Table 1. 13th Generation Intel® Core™ S-Series Processors with Intel® R680E Platform Controller Hub (PCH) – Corporate/Mainstream

	i9-13900E	i7-13700E	i5-13500E	i513400E	i313100E
Use Condition	Embedded	Embedded	Embedded	Embedded	Embedded
Cores[1]	32 (8 + 16)	24 (8 + 8)	20 (6 + 8)	16 (6 + 4)	8 (4 + 0)
Intel® TCC		Yes	Yes	Yes	Yes

Table 2. 13th Generation Intel® Core™ S-Series Processors with Intel® R680E Platform Controller Hub (PCH) – Low Power

	i9-13900TE	i7-13700TE	i5-13500TE	i3-13100TE
Use Condition	Embedded	Embedded	Embedded	Embedded
Cores[1]	32 (8 + 16)	24 (8 + 8)	20 (6 + 8)	8 (4 + 0)
Intel® TCC		Yes	Yes	Yes

Table 3. 13th Generation Intel® Core™ U-Series Processors (15W)

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

Table 4. 13th Generation Intel® Core™ P-Series Processors (28W)

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

Table 5. 13th Generation Intel® Core™ H-Series Processors (45W)

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

Note: For more details on individual SKU configurations, contact your Intel representative.

2.2.1.1 Customer Reference Boards (CRBs)

13th Generation Intel® Core™ Processors reference platform is available for evaluation and development.

2.2.1.2 Intel® 600 Series Chipset Family Platform Controller Hub (PCH)-S R680E SKU

13th Generation Intel® Core™ S-Series Processor is combined with Intel® 600 Series Chipset Family PCH-S to complete the platform. This section and its subsections reference the TSN capable Ethernet controllers. For Intel® Core™ S-Series processors only the R680E SKU of the PCH support the integrated 2.5 Gb Ethernet with TSN technology.

2.2.1.2.1 TSN-capable Ethernet Controllers

The 13th Generation Intel® Core™ Processor family support TSN over Ethernet via Intel's discrete i226-LM/IT Ethernet controllers and via the 13th Generation Intel® Core™ Processors Platform (R680E SKU of PCH only for Intel® Core™ S-Series Processors only) which contains a total of three of the following integrated ethernet ports for Intel® Core™ S-Series Processors and 2 for Intel® Core™ P-Series Processors:

Port	Number of Ports	Space	Media-Access Control (MAC)
Integrated Ethernet	1	1 GbE with vPro support	Yes
Integrated Ethernet with TSN support	• 2 (S-Series via PCH) • 1 (U/P/H-Series)	2.5 GbE with TSN support (for each port)	Yes (for each port)

The Ethernet and Ethernet with TSN controllers are physically isolated from each other and are designed to run concurrently.

This chapter is focused on the behavior of the integrated 2.5 Gb Ethernet controllers with TSN support. TSN is a set of IEEE standards that are intended to ensure quality transmission of the time sensitive data over Ethernet networks. The Ethernet-TSN controllers can operate at multiple speeds: 10 Mbps, 100 Mbps, 1 Gbps, and 2.5 Gbps (SGMII) and in either full duplex or half duplex mode.

The Ethernet-TSN MAC is accessed by the 13th Generation Intel® Core™ 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.

This MAC has a Station Management (STA) Entity that is accessible to software via the Memory Map IO (MMIO) registers to control the associated MDIO interface. Refer to the following:

- IEEE Standard 802.3, Clause 22, and Clause 45. The Physical Coding Sublayer (PCS) module provides the sublayer circuitry between the GMII of the MAC and Ten-Bit Interface (TBI) of the Serial Gigabit Media-Independent Interface (SGMII) circuitry.
- IEEE Standard 802.3 Clause 35 for GMII and Clause 36 for TBI.

To support TSN standards, the chipset's two integrated Ethernet-TSN controllers require Ethernet PHYs external of the chipset.

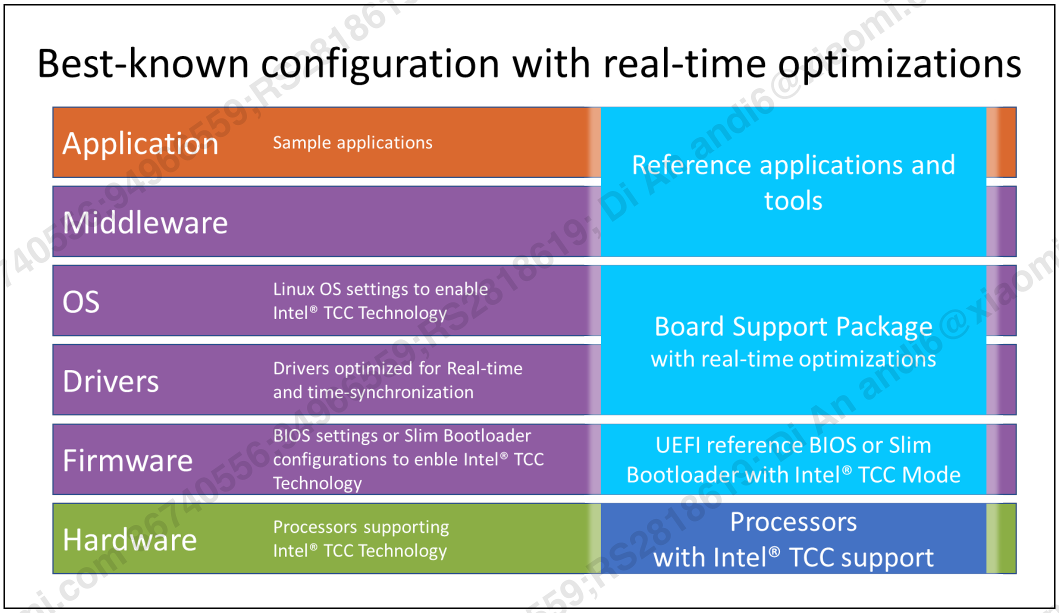
2.2.2 Where to Get Hardware

Contact your Intel representative to acquire hardware.

2.3 Intel® TCC Software

Intel provides software that enables developers to access Intel® TCC hardware features, including BIOS support, Linux* kernel drivers and patches, and sample applications.

Figure 1. Intel® TCC Reference Software Stack


Best-known configuration with real-time optimizations
Application
Sample applications
Middleware
Reference applications and tools
OS
Linux OS settings to enable Intel® TCC Technology
Drivers
Drivers optimized for Real-time and time-synchronization
Board Support Package with real-time optimizations
Firmware
BIOS settings or Slim Bootloader configurations to enble Intel® TCC Technology
UEFI reference BIOS or Slim Bootloader with Intel® TCC Mode
Hardware
Processors supporting Intel® TCC Technology
Processors with Intel® TCC support

2.3.1 Application and Middleware Support

Intel provides application and middleware support, including tools and sample applications that enable developers to use certain Intel® TCC and TSN features (described later in this document).

2.3.2 Operating System Support

Intel provides Linux* OS support in the form of the Yocto Project*-based board support packages (BSP) and Ubuntu with Kernel Overlay. The BSP contains real-time configuration settings and the Linux PREEMPT_RT patch. The configuration is pulled in from the upstream branch by the Yocto/Ubuntu Project build to the BSP. The BSP is part of the BKC. For information about the BKC, see Best Known Configuration (BKC).

Support for other operating systems is provided by their respective vendors.

This document describes OS tuning in more detail, as part of the Intel® TCC Platform Tuning Strategy.

2.3.3 Driver Support

Intel provides Linux OS driver support with its Yocto Project-based BSP and Ubuntu with Kernel Overlay. The BSP-provided drivers implement support for Linux APIs, enabling software coordination of time between locally connected (i.e. PCIe*) and integrated devices using dedicated time synchronization hardware. Devices with support for hardware-based time coordination capabilities are:

- Precision Time Measurement PTM-capable PCIe devices
- PTM-capable TSN-enabled PCIe Ethernet network devices
- Time Aware GPIO

The BSP-provided drivers use virtual channels to prioritize real-time data flows within the platform to the Ethernet interface where available.

2.3.4 BIOS Support

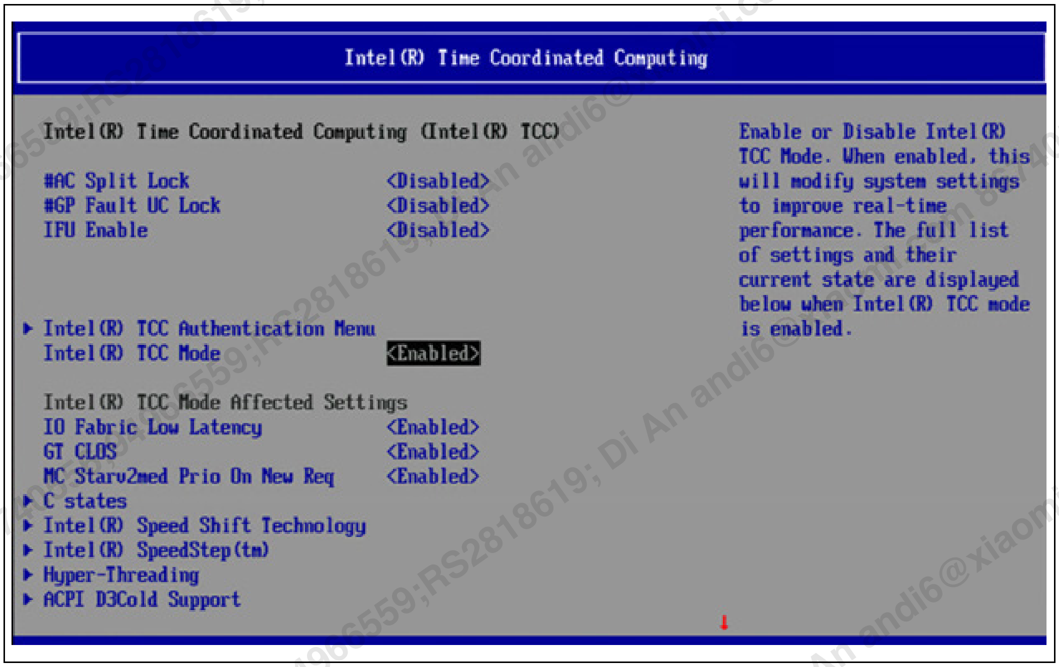
Many of the control registers involved in tuning are accessible only through the BIOS. Some companies may not want to enable real-time tuning as it fundamentally changes the performance profile of the platform. Real-time tuning requires configurability, which can be modified without recompiling and reflashing the BIOS.

Intel provides a reference BIOS that enables companies to control tuning through an option called Intel® Time Coordinated Computing Mode (Intel® TCC Mode). This option is in the following 13th Generation Intel® Core™ Processors BIOS menu:

EDKII Menu > Platform Configuration > Intel® Time Coordinated Computing > Intel® TCC Mode

Figure 2 is an example of the Intel® TCC Mode interface in the reference BIOS. The interface may differ depending on a given BIOS version.

Figure 2. Example of Intel® TCC Mode Interface



```
                      Intel(R) Time Coordinated Computing

  Intel(R) Time Coordinated Computing (Intel(R) TCC)            Enable or Disable Intel(R)
                                                                TCC Mode. When enabled, this
  #AC Split Lock                 <Disabled>                     will modify system settings
  #GP Fault UC Lock              <Disabled>                     to improve real-time
  IFU Enable                     <Disabled>                     performance. The full list
                                                                of settings and their
                                                                current state are displayed
                                                                below when Intel(R) TCC mode
▸ Intel(R) TCC Authentication Menu                              is enabled.
  Intel(R) TCC Mode              <Enabled>

  Intel(R) TCC Mode Affected Settings
  IO Fabric Low Latency          <Enabled>
  GT CLOS                        <Enabled>
  MC Starv2med Prio On New Req   <Enabled>
▸ C states
▸ Intel(R) Speed Shift Technology
▸ Intel(R) SpeedStep(tm)
▸ Hyper-Threading
▸ ACPI D3Cold Support
                                                  ↓
```

Intel® TCC Mode provides access to a collection of settings. Some of these settings appear only in the Intel® TCC submenu, while others are shortcuts to settings located elsewhere in the BIOS.

When Intel® TCC Mode is enabled, it configures the settings to predetermined values, however, those values can be changed independently. We recommend not deviating from the default Intel® TCC Mode values to ensure consistent configuration that is optimized for general real-time performance.

When Intel® TCC Mode is disabled, the settings can be set independently for granular control of specific capabilities. By selecting one of these settings from the Intel® TCC submenu, the BIOS jumps to the BIOS page where each setting resides.

Intel® TCC Mode affected settings fall into the following categories:

- Enabling certain Intel® TCC hardware capabilities.
- Disabling certain power management capabilities:
  - Hardware and software power management features can negatively affect real-time performance for various I/O paths.
  - Intel® TCC Mode disables power management settings in the BIOS.

This document describes these settings in more detail, as part of the Intel® TCC Platform Tuning Strategy.

2.3.4.1 Additional Information

The reference BIOS is part of the BKC. For information about the BKC, see Best Known Configuration (BKC).

For register specifications, see the *BIOS Specification* (formerly BIOS Writer's Guide) listed in Reference Documents.

2.3.5 Slim Bootloader Support

Like the BIOS, the Slim Bootloader (SBL) provides a configuration option to enable or disable Intel® TCC Mode. 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 Intel® TCC Mode, data streams optimizer, Software SRAM, and Error logging using the ConfigEditor tool. For more information on updating the settings using this tool, refer to the SBL wiki:

https://slimbootloader.github.io/how-tos/enable-intel-tcc.html#open-sbl-defaultconfiguration-data

For generic ConfigEditor info, refer to:

https://slimbootloader.github.io/tools/ConfigTools.html?highlight=configeditor#configeditor

2.4 Other Analysis and Profiling Tools

Intel provides the following tools to help developers detect and analyze areas of the system negatively affecting real-time performance.

Table 6. Other Analysis and Profiling Tools

Tool	Description
Intel® VTune™ Profiler	Detect and analyze sources of system jitter. For information specific to real-time applications, see *Tutorial: Profile Applications with Intel® VTune™ Profiler* listed in Reference Documents.
Inter-event histogram triggers tool	A tool in the kernel that can assist with latency profiling and analysis. This tool is platform independent. See Section 2.2 of the kernel document *Event Histograms* in Reference Documents.

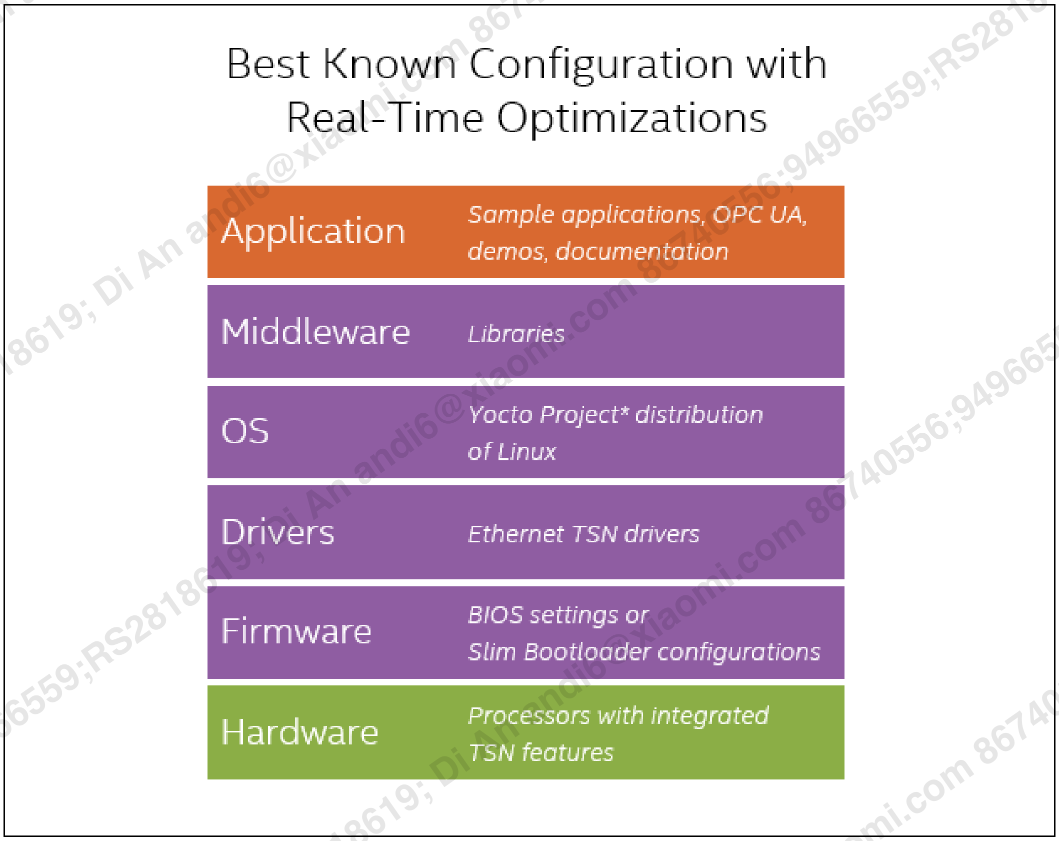
Note: These tools do not currently support 13th Generation Intel® Core™ Processors. Contact your Intel representative for details.

2.5 Time-Sensitive Networking Software

Intel provides software that enables TSN features, including the following:

- Ethernet drivers
- OS support
- Open-source libraries and utilities (open62541, iproute2, ethtool, linuxptp)
- Sample applications (socket-level and OPC UA Publish/Subscribe over TSN).

Figure 3. TSN Software



Best Known Configuration with Real-Time Optimizations

Application — *Sample applications, OPC UA, demos, documentation*

Middleware — *Libraries*

OS — *Yocto Project* distribution of Linux*

Drivers — *Ethernet TSN drivers*

Firmware — *BIOS settings or Slim Bootloader configurations*

Hardware — *Processors with integrated TSN features*

The Yocto Project-based BSP and Ubuntu with Kernel Overlay listed in Best Known Configuration (BKC) provides all TSN Software.

2.6 Best Known Configuration (BKC)

The BKC consists of the following components:

- Reference Unified Extensible Firmware Interface (UEFI) integrated firmware image (IFWI) or slim bootloader (SBL)
- Drivers
- OS:
  - Yocto Project-based Board Support Package for Raptor Lake – S for IoT Platforms
  - Ubuntu with Kernel Overlay for Raptor Lake – P for Edge Platforms

Intel real-time KPIs are based on the combination of a supported processor, the BKC, and the provided Intel workload.

Note: The BKC is tuned for the KPI workload and may not be the best configuration for other workloads possibly requiring additional tuning.

You can run the Intel workload with the processor and BKC to replicate the KPIs in your lab to start assessing the effects of BKC components on performance.

The BKC configuration relies on the recommended BIOS with Intel® TCC Mode enabled and kernel configuration in Sections 2.3.4 and 6.1, respectively. The kernel configuration is reflected in the board support package. However, the "Intel® TCC Mode" option in the BIOS must be set to enable, to replicate KPI performance.

For more information about the BKC, see in Reference Documents:

- Section 10.0 for Yocto Project-based Board Support Package for Raptor Lake – S for IoT Platforms - Release Notes (Beta Release).
- Ubuntu with Kernel Overlay for Raptor Lake – P for Edge Platforms (Release Note).

More information on KPIs is detailed in the following Real-Time Key Performance Indicators (KPIs).

§

3.0 Real-Time Key Performance Indicators (KPIs)

Key performance indicators (KPIs) are performance metrics. Intel provides information about KPIs, including measurements and configurations for replicating the tests on a given platform, in a later Gold Deck release (see Reference Documents).

The Performance Report describes:

- Real-time capabilities enabled at product launch of the industrial processors.
- Real-time KPI requirements for use of Intel BKC / board support package (BSP) and associated documentation.

The information is expected to cover the following two benchmarks:

- **Cyclic Test**: a latency test designed to measure a system's real-time performance and event responsiveness. Cyclic test's latency is defined as the time between a thread's intended wake-up time and its actual wake-up time.
- **Real-Time Compute Performance (RTCP):** a KPI developed by Intel measuring the total elapsed time from when a synthetic sensor data packet is received, to when a synthetic actuator command packet is sent (including the elapsed time for the execution of a predefined compute workload).

§

4.0 Intel® TCC Platform Tuning Strategy

Intel® Time Coordinated Computing (Intel® TCC) tuning is the primary responsibility of the Intel® TCC Mode BIOS option, and OS configuration. Intel systems are meant for multiple use cases and the changes in BIOS configurations are meant to tune for specific use cases. The BIOS is usually configured for a default non-Real-Time use case.

Thus, a real-time BKC exists for real-time tuning to achieve peak performance based on a reference use case and customers may need further tuning to reach their desired performance for their specific use cases. This tuning is divided into the following categories:

- System software tuning
- Platform power management tuning
- Intel® TCC features tuning
- Fabric tuning

Different types of tuning have different effects on latency. *Latency* refers to the duration of time between two events. It can mean any type of latency: I/O latency, buffer access latency, and the sum of these and others—network *packet in* to *network packet out* (cycle time).

Usages with more relaxed requirements need less tuning to meet their latency requirements, while usages with stricter requirements necessitate substantial effort in tuning. When tuning the platform, address higher-impact tuning first, then work down to lower-impact tuning until the KPI requirements are satisfied.

For example, taking an out-of-the-box system and performing system software tuning can reduce latency on the order of milliseconds and decrease latency by a significant improvement. After performing system software tuning, if requirements still are not met, you can implement platform power management tuning.

Power management tuning can reduce latency to hundreds of microseconds, which is much less than system software tuning in absolute terms. Power Management tuning can also further reduce the latency jitter by hundreds of microseconds, complementing the software tuning that has been implemented.

System Software Tuning: System software has the highest impact on real-time performance and can increase worst-case execution time (WCET) by many orders of magnitude. Tune system software such as OS and drivers before even attempting to begin optimizing hardware features.

The BKC recommends various system software tuning settings such as Linux kernel build configuration settings (for example, “`CONFIG_PREEMPT_RT=y`”) and boot parameters (for example, isolcpus).

Platform Power Management Tuning: After system software has been tuned, move on to the tuning platform for power management. Platform power management often negatively affects real-time performance by putting the platform and subcomponents into low-power states or by throttling performance (for example, via frequency scaling).

The BKC recommends various platform power management tuning settings such as the Linux kernel boot parameters (for example, `intel_idle.max_cstate=0`).

Intel® TCC Features Tuning: After system software and power management have been tuned, the system is running at the optimal hardware performance using the basic platform capabilities. To further improve real-time performance, the platform provides a set of Intel® TCC features. Intel® TCC features tune targets' specific transaction flows, compute workloads, or corner cases.

Fabric Tuning: *Fabric* refers to the interconnect technology that carries on-chip communications between the different functional components of the processor. Fabric tuning includes microarchitecture-specific optimizations for the functional components in the fabric that include prioritization, arbitration, and flow control. These tunings come at a high cost in implementation time and performance tradeoffs, and they provide comparatively little improvement relative to the other types of tuning. However, the best possible real-time performance can be attained by tuning the fabric arbitration and prioritization characteristics (though only after the other three types of tuning have been completed).

§

5.0 System Software Tuning

System software has the highest impact on real-time performance and can improve worst-case execution time (WCET) by many orders of magnitude. This can take the form of various sources of jitter including, but not limited to, scheduler preemption, Read-Copy-Update (RCU) callbacks, and poorly optimized drivers.

Intel recommends tuning system software such as OS and drivers before attempting to optimize hardware features. Accordingly, a good method for debugging massive latency spikes (higher than 1 millisecond) is to first ensure the OS (and higher-level software) has been sufficiently tuned. Scheduler-caused latency spikes also tend to exhibit periodicity.

Tuning system software for real-time often comes at the cost of average performance and multitasking, reducing the maximum aggregate throughput of the system.

This section describes the configurations and settings that may be implemented to tune a given platform to improve its real-time performance. It includes OS configurations and kernel parameters. It also includes what libraries are required, the command-line optimizations during kernel boot time, and OS behavior changes to improve platform performance.

5.1 Linux Operating Systems

5.1.1 Kernel Parameters

This section explains the kernel parameters that you can configure to tune the performance of the platform. The parameters are based on kernel version 5.10 with the PREEMPT_RT patch.

In the BKC, the kernel is compiled with the following parameters. You can find a description of each parameter in the specified kernel configuration file.

Table 7. Other Analysis and Profiling Tools

Kernel Build Parameter	Value	See Kernel Config File for Description
`CONFIG_GENERIC_IRQ_MIGRATION`	Y	`kernel/irq/Kconfig`
`CONFIG_HIGH_RES_TIMERS`	Y	`kernel/time/Kconfig`
`CONFIG_CPU_ISOLATION`	Y	`init/Kconfig`
`CONFIG_RCU_NOCB_CPU`	Y	`kernel/rcu/Kconfig`

Kernel Build Parameter	Value	See Kernel Config File for Description
CONFIG_SMP	Y	arch/x86/Kconfig
CONFIG_MIGRATION	Y	mm/Kconfig
CONFIG_PCIEPORTBUS	Y	drivers/pci/pcie/Kconfig
CONFIG_PCIE_PTM	Y	drivers/pci/pcie/Kconfig
CONFIG_GENERIC_IRQ_MIGRATION	Y	kernel/irq/Kconfig
CONFIG_EXPERT	Y	init/Kconfig
CONFIG_PREEMPT_RT	Y	kernel/Kconfig.preempt ***Note:*** This option is only used for preempt kernels and not required for tuning on standard kernels. Documentation can be found in the real-time kernel's source.
CONFIG_PREEMPT_RT_FULL	Y	kernel/Kconfig.preempt ***Note:*** This option is only used for preempt kernels and not required for tuning on standard kernels. Documentation can be found in the real-time kernel's source.
CONFIG_CPU_FREQ	Y	drivers/cpufreq/Kconfig ***Note:***.This CONFIG flag is enabled since it has dependencies on other CONFIG flags.
CONFIG_SCHED_MC_PRIO	N	arch/x86/Kconfig
CONFIG_PREEMPT_RCU	Y	kernel/rcu/Kconfig
CONFIG_HUGETLBFS	Y	fs/Kconfig
CONFIG_EFI	Y	arch/x86/Kconfig

The kernel should be booted with the following parameters. For descriptions of these parameters, see Linux kernel command-line parameters in Reference Documents.

Table 8. Kernel Command-Line Parameters

Kernel Command-Line Parameter	Value
i915.enable_guc	7
processor.max_cstate	0
processor_idle.max_cstate	0

Kernel Command-Line Parameter	Value
intel.max_cstate	0
intel_idle.max_cstate	0
clocksource	tsc
tsc	reliable
nowatchdog	No value required. ***Note***: Option may only benefit performance of some real-time workloads. Recommended on a case-by-case basis.
intel_pstate	disable
idle	poll
noht	No value required. This can cause the timer tick to be enabled even with nohz_full set.
isolcpus	X-Y (where X is the first CPU and Y is the last CPU in a consecutive list of CPUs allocated to real-time applications, for example, 2-3)
rcu_nocbs	X-Y (where X is the first CPU and Y is the last CPU in a consecutive list of CPUs allocated to real-time applications, for example, 2-3)
rcu_nocb_poll	No value required
irqaffinity	X-Y (where X is the first CPU and Y is the last CPU in a consecutive list of CPUs allocated to handling interrupts)
cpufreq.off	1 ***Note***: Option may not be applicable to all kernel versions and may only benefit performance of some real-time workloads. On some kernel versions, it may be appropriate to handle after boot to ensure frequency is locked at max.
nosoftlockup	No value required
rcupdate.rcu_cpu_stall_suppress	1 ***Note***: Option may only benefit performance of some real-time workloads. Recommended on a case-by-case basis.
rcu_nocb_poll irqaffinity	0
mce	off
hpet	disable

Kernel Command-Line Parameter	Value
numa_balancing	disable
igb.blacklist	no
nohz	No value required. ***Note:*** Although nohz is a common real-time parameter, Intel has found that nohz negatively affects the real-time performance of this processor. Intel recommends omitting nohz in the kernel boot parameters for this processor
nohz_full	X-Y (where X is the first CPU and Y is the last CPU in a consecutive list of CPUs allocated to real-time applications) ***Note:*** Although nohz_full is a common real-time parameter, Intel has found that nohz_full negatively affects the real-time performance of this processor. Intel recommends omitting nohz_full in the kernel boot parameters for this processor.
efi	runtime
iommu	pt ***Note***: Option may only benefit performance of some real-time workloads. Recommended on a case-by-case basis.
art	virtallow ***Note***: Option may only benefit performance of some real-time workloads. Recommended on a case-by-case basis.
i915.enable_rc6	0 ***Note***: Option may not be applicable to all kernel versions and may only benefit performance of some real-time workloads.
nmi_watchdog	0
nosoftlockup	No value required.
hugepages	1024

Variant configurations can be found in Appendix A.

5.1.2 Other Linux OS Optimizations

This section provides additional optimizations that can be done in the OS environment to improve real-time performance. These optimizations are:

- Changing the affinity of a certain process to align on a specific CPU
- Modifying the behavior of the scheduler
- Isolating certain cores for certain tasks

Modifying the scheduler to prioritize some tasks over others can help reduce downtime, which helps to achieve real-time performance. Linux tools, such as taskset and numactl, enable the binding of a process or thread to core. This can help localize memory and reducing memory accesses, leading to more desirable performance. Another tool that can be used to modify scheduling attributes is chrt, which is a tool used to manipulate the real-time attributes of a process.

Caution*: Enabling chrt to manipulate the scheduler may cause conflicts with isolcpus settings.*

Numactl Example:

```
numactl  --cpunodebind=0 test.sh
```

This example binds all processes generated by test.sh to cores found only in CPU0.

By running this command, the processes generated by test.sh will have access to localized memory, resulting in better latency results than if processes were on separate CPU nodes.

CHRT Example:

```
chrt -f 99
```

This example sets the scheduling policy of PID 99 to *SCHED_FIFO*. To find the PID, run `pidof -s` **[process_name]**

Taskset Example:

```
taskset -c 1,2,3 test.sh
```

This example assigns the processes generated to be bound to cores 1-3, isolating test.sh from interference from processes running on core 0.

By running this command, the processes generated by test.sh will have access to localized memory, resulting in better latency results than if processes were on separate CPU nodes.

Shared Memory Device Example:

```
taskset -c 3 tests.sh > /dev/shm/output.log
```

When outputting data from a real-time core, consider outputting to memory rather than disk or graphics. Transactions to memory have a much better performance profile than using disk or graphics.

```
taskset -c 1 copy.sh /dev/shm/output.log /home/root/output.log
```

If persistent data storage is necessary, data can be moved from memory to disk using other cores.

Fixed Max Frequency Example:

```
cat /sys/devices/system/cpu/cpu3/cpufreq/scaling_max_freq >
/sys/devices/system/cpu/cpu3/cpufreq/scaling_min_freq
```

This example sets the minimum CPU frequency on cpu3 to the maximum CPU frequency on cpu3.

When Intel SpeedStep® and Intel® Speed Shift Technology are enabled in the BIOS, Intel recommends you to use the Linux OS infrastructure to set min=max frequency for the real-time cores.

5.1.3 Linux OS Resource Partitioning

The goal of this section is to provide insight into a methodology for partitioning resources on a platform using OS level tools.

The broad methodology relies on sectioning off cores for real-time workloads and leaving one core for best-effort work and system management. Typically, this is achieved with a combination of the command-line parameter *isolcpus* and the runtime parameter *taskset*. Isolcpus instructs Linux to not schedule anything on the instructed cores — they are considered isolated from the scheduler. Taskset then enables you to lock a program to a core. If the chosen core is one of the isolated cores, that program will be the only thing to run on that core. For example, in a four-core system, you could isolate cores 1 through 3 and leave core 0 for system management, then taskset a program to run on core 1.

Unfortunately, these tools do not restrict Linux from running certain kernel threads on the core, including interrupt handling. A couple methods to minimize the impact of these system management threads are moving interrupts and deprioritizing system threads.

To move interrupts, use the /proc/irq filesystem options in Linux. You can use the smp_affinity option for each registered interrupt to affine the interrupt to a specific core. For any interrupts that are not used by the real-time application, use the smp_affinity option to set them to the best-effort core. Make sure the interrupts that are important to the real-time application remain on the real-time core.

Furthermore, using the PREEMPT_RT patch for Linux will expose system kernel threads as processes — using the function chrt enables you to set the priority and scheduling method of these processes. It is recommended to set both the system threads to a low priority and the critical workload priority much higher. Typically, a few processes are exposed such as ksoftirq (these can be seen by running ps –e). The core number assigned to a process appears at the end of the process name, for example, ksoftirq/2. Additionally, RCU processes are exposed as well. A few different types are available, but they also follow the same nomenclature of having a /<core> at the end of the process name. Using chrt, deprioritize these (priority 0), and set the scheduling method of these to SCHED_OTHER. SCHED_FIFO and a higher priority should be used for the critical workload.

Commands of Interest with Examples

Note: Based on the real-time workload, isolcpus may need to be changed to provide more best-effort cores for the kernel.

1. Command-Line Parameters:

```
processor.max_cstate=0
intel.max_cstate=0
processor_idle.max_cstate=0
intel_idle.max_cstate=0
clocksource=tsc
tsc=reliable
nowatchdog
intel_pstate=disable
idle=poll
noht
isolcpus=2,3 (offloads residual 1Hz scheduler tick)
rcupdate.rcu_cpu_stall_suppress=1
rcu_nocb_poll
rcu_nocbs=all (offloads RCU callbacks)
irqaffinity=0
i915.force_probe=*
i915.enable_guc=7
i915.enable_rc6=0
i915.enable_dc=0
i915.disable_power_well=0
hugepages=1024
mce=off
hpet=disable
numa_balancing=disable
igb.blacklist=no
efi=runtime
art=virtallow
iommu=pt
nmi_watchdog=0
nosoftlockup
nohz_full=2-3 (sets the scheduler clock to 1Hz if only one user-
space app is running)
```

2. OS Tools:

```
taskset -c <core> <application>
chrt <scheduler method> -p <priority> <application PID>
       chrt -f -p 98 <application PID> = FIFO scheduling with
execution priority 98 (of 99)
       chrt -o -p 0 <application PID> = OTHER scheduling with
execution priority 0
/proc/irq/<irq number>/smp_affinity = core mask for affined cores
```

§

6.0 Platform Power Management Tuning

After System Software Tuning, real-time performance can be further improved by tuning platform power management. Platform power management affects real-time performance by putting the platform and subcomponents into low-power states or by throttling performance (for example, via frequency scaling). Entering and exiting these low-power states may incur significant latency penalties; similarly, frequency throttling or transitions can degrade performance as well.

High latency spikes caused by power management are characterized by their tendency to appear when the system is idle but disappear when the system is loaded with traffic. Related to power management, but rarely observed, is throttling caused by thermal events that often appear as a sudden drop in average (and worst) case performance after an extended runtime.

Platform power management tuning often involves disabling power management; thus, a tradeoff between real-time performance and TDP is required. BIOS largely handles power management tuning, but some power management is best or necessarily handled at the OS or driver level.

This section delves deeper into the power management (PM) aspect of the system that affects real-time performance, including the system PM, and the I/O fabric PM.

6.1 Intel® TCC Mode (BIOS): Power Management Settings

When enabled in BIOS, Intel® Time Coordinated Computing Mode (Intel® TCC Mode) sets the following power management options.

Table 9. Intel® TCC Mode (BIOS): Power Management Settings

Setting Name	Option	Menu	Description
Intel® Data Direct I/O Technology (Intel® DDIO)	Enabled	Intel Advanced Menu\System Agent (SA) Configuration	When enabled, turns on Intel® Data Direct I/O Technology (Intel® DDIO).
Enable Monitor MWAIT	Disabled	Socket Configuration\CPU C State Control	Allows monitor and MWAIT instructions. Disabling will prohibit the system from requested C-states.
DRAM RAPL	Disabled	Socket Configuration\ Advanced Power Management Configuration \ Memory Power & Thermal Configuration\ DRAM RAPL Configuration	RAPL: Running Average Power Limiting technology

Setting Name	Option	Menu	Description
R-link ASPM Support	Disabled	Socket Configuration\ IIO Configuration\ R-Link	This option can disable ASPM support in a PCIe root port. "Auto" Keeps the hardware default
Hyper-Threading	Disabled	Socket Configuration\ Processor Configuration	
Intel SpeedStep® (P-States)	Disabled	Socket Configuration\ Advanced Power Management Configuration\ CPU P-State Control	When *disabled*, turns off Intel SpeedStep® technology. The technology enables the management of processor power consumption via performance state (P-State) transitions.
Hardware P-states Option: Disabled	Disabled	Socket Configuration\ Advance Power management \Hardware PM State Control	This option can disable P-state calls based on OS request
Page Policy	Adaptive	Socket Configuration\ Memory Configuration\ Page Policy	Set memory page policy parameters
CKE Throttling	Manual	Socket Configuration\ Advance Power management Configuration\ Memory Power and Thermal Configuration\ Memory Power Savings Advanced Options\	Configures CKE (DDR Clock Enable) Throttling
APD	Disabled	Socket Configuration\ Advance Power management Configuration\ Memory Power and Thermal Configuration\ Memory Power Savings Advanced Options\ CKE Feature	Active Power Down
PPD	Disabled	Socket Configuration\ Advance Power management Configuration\ Memory Power and Thermal Configuration\ Memory Power Savings Advanced Options\ CKE Feature	Pre-Charged Power Down

Setting Name	Option	Menu	Description
Platform Control Hub (PCH) PCI Express Configuration	For each root port: • ASPM: Disabled • L1 Substates: Disabled • PTM Enabled • Multi Virtual Channel (MVC)	Platform Configuration\ PCH-IO Configuration\ PCI Express Configurations\ PCI Express Root Port 0-12 Platform Configuration\ PCH-IO Configuration\ TSN GBE Configuration	**ASPM**: When disabled, turns off Active State Power Management (ASPM). ASPM is an autonomous hardware-based, active state mechanism that enables power savings even when the connected components are in the D0 state. After a period of idle link time, an ASPM Physical-Layer protocol places the idle link into a lower power state. **L1 Substates**: When disabled, turns off PCIe L1 substates. The fundamental idea behind L1 substates is to use something other than the high-speed logic inside the PCIe transceivers to wake the devices. The goal is to achieve near zero power consumption with an active state. **PTM**: When enabled allows Precision Time Measurement between two devices via time stamping capability. **MVC**: When enabled allows VC1 Traffic for real time and VC0 for best effort traffic.
Legacy IO Low Latency	Enabled	Platform Configuration\ PCH-IO Configuration	Set to enable low latency of legacy IO. Some system required lower IO latency irrespective of power. This is a tradeoff between power and IO latency.
PCH LAN: Fast Startup Wake Support	Disabled	Platform Configuration\ PCH-IO Configuration\Intel Test Menu\PCH Advanced Menu	Wake on LAN enabled or disabled when Fast Startup is enabled
R-link ASPM Enabled	Disabled	Platform Configuration\ PCH-IO Configuration\ Debug	Disabled L1 ASPM for the R-link

Setting Name	Option	Menu	Description
CPU PCI Express Configuration: Port 1A and R-link	For each root port: • ASPM: Disabled • L1 Substates: Disabled	Socket Configurations\ IIO Configurations\ Socket 0 Configuration\ Port 1A or R-link	**ASPM**: When *disabled*, turns off Active State Power Management (ASPM). ASPM is an autonomous hardware-based, active state mechanism that enables power savings even when the connected components are in the D0 state. After a period of idle link time, an ASPM Physical-Layer protocol places the idle link into a lower power state. **L1 Substates**: When *disabled*, turns off PCIe L1 substates. The fundamental idea of L1 substates is achieving near zero power consumption with an active state—without using the high-speed logic inside the PCIe transceivers to wake the device.
RC6 (Render Standby)	Disabled	Intel Advanced Menu\Power & Performance\GT – Power Management Control\	Disabled Render Standby Support

6.2 System PM

6.2.1 Intel® Turbo Boost Technology (P0 States)

Intel® Turbo Boost Technology uses the principle of leveraging thermal headroom to dynamically increase processor performance for single-threaded and multi-threaded/multi-tasking environment. P0 states are also known as turbo states or P0 frequency is usually referred to as turbo frequency.

Intel® Turbo Boost Technology is set to **disabled**.

6.2.2 Enhanced Intel SpeedStep® Technology and Intel SpeedStep® (P-States)

P-states are the various execution power states of the processor. These are the frequency-voltage pairs that dictate the speed at which the processor will run.

Conventional Intel SpeedStep® Technology switches both voltage and frequency in tandem between high and low levels in response to processor load. A later version of Intel SpeedStep® called Enhanced Intel SpeedStep® Technology was released. It is an advanced means of enabling high performance while meeting the power-conservation needs of mobile systems. Enhanced Intel SpeedStep® Technology builds upon that architecture using design strategies such as Separation between Voltage and Frequency Changes, and Clock Partitioning and Recovery.

These two items are generally known as Intel SpeedStep® and is set to disabled, as P-states transitions introduce latencies.

6.2.3 C-States

C-states are various idle states the individual cores can be in and are meant to save power while the processor or cores are not executing any instructions. There are many C-states ranging from C0 where the CPU is fully turned on to C6 where the CPU internal voltages are reduced to any value, including 0 V (also referred to as deep power down).

C-states are disabled for real-time configurations as transitions between C-states introduce latencies.

6.2.4 S-States

S-states are sleep states of a system ranging from S0, the most demanding S-state, to Sx, where x > 0, which could range from sleep to hibernate.

S-states are disabled for real-time configurations as transitions between S-states introduce latencies.

6.2.5 Display C-States

Display Engine provides internal power states that place the component into low-power modes of operation (DC0–DC9) that are initiated during Package C-State flows. For example, the system cannot enter package C10 unless Display C-States are enabled.

6.3 I/O PM

There are several stages of PM in the I/O system. The fabric where multiple I/O units are switched can have its own PM. The individual I/O units can have its own PM as well.

6.3.1 I/O Fabric PM

The I/O system can have one or more stages of I/O switch fabric. Each of these will likely have its PM unit to manage power consumption more efficiently.

6.3.2 Individual I/O System PM

Currently, the I/O system that is of importance to real-time performance is the Peripheral Component Interconnect express* (PCIe*) interface. Intel® TCC/TSN-capable devices are currently plugged into the PCIe expansion slots. The power management mechanism for PCIe is called ASPM, L1, etc. Currently, to get the best real-time performance out of the system, these are generally disabled in BIOS.

7.0 Intel® TCC Features Tuning

After system software and power management have been tuned, the system is running at the optimal hardware performance using the basic platform capabilities. To further improve real-time performance, the platform provides a set of Intel® Time Coordinated Computing (Intel® TCC) features.

Intel® TCC feature tuning targets specific transaction flows, compute workloads, or corner cases. This tuning often requires a tradeoff of performance between the WCET of critical real-time paths, and best-effort workloads or average execution time (throughput).

Intel® TCC Mode configures relevant Intel® TCC features to predetermined values that have shown to provide the best general boost to real-time performance, though the best performance for a specific workload may require independent configuration of Intel® TCC features.

7.1 Intel® TCC Mode (BIOS): Intel® TCC Feature Settings

7.1.1 Overview

The BIOS provides an option that configures many individual settings in a single location. These settings include:

- Disabling features that are known to negatively impact latency/jitter
- Enabling features known to increase temporal isolation
- Configuring the platform to bias more toward lower latency versus throughput or power efficiency

7.1.2 Architecture

Some notable features affected by Intel® TCC Mode include the following (details of these features are covered below):

- Cache allocation
- PCIe/fabric virtual channels

7.1.3 Usage

Customers who desire the increased temporal isolation and low latency afforded by this setting and are willing to accept the trade-off in terms of reduced best-effort performance and increased power consumption, would enable this capability in the BIOS.

7.1.4 Intel® TCC Mode (BIOS): Intel® TCC Feature Settings

When enabled, Intel® TCC Mode sets the following Intel® TCC options:

Table 10. Intel® TCC Mode (BIOS): Intel® TCC Feature Settings

Setting Name	Option	Menu	Description
CPU PCI Express Configuration	For each root port: • PTM: Enabled • Virtual Channels (VC): Enabled Multi-VC: Enabled	**PTM**: Intel Advanced Menu\System Agent (SA) Configuration\PCI Express Configuration\PCI Express Port # **VC**: Intel Advanced Menu\System Agent (SA) Configuration\PCI Express Configuration\PCI Express Port # **Multi-VC**: Intel Advanced Menu\System Agent (SA) Configuration\PCI Express Configuration\PCI Express Port #	**PTM**: When enabled, provides a hardware mechanism for PCIe devices to correlate clock domains between an endpoint and the root complex. **VC**: When enabled, turns on PCIe virtual channel capability. PCIe virtual channels provide a mechanism for differentiating PCIe traffic throughout the fabric. **Multi-VC**: When enabled, this provides support for multiple PCIe virtual channels (13th Generation Intel® Core™ Processors supports 2 VCs).
PCH PCI Express Configuration	For each root port: PTM: Enabled	**PTM**: Intel Advanced Menu\PCH-IO Configuration\PCI Express Configuration\PCI Express Root Port #	**PTM**: When enabled, provides a hardware mechanism for PCIe devices to correlate clock domains between an endpoint and the root complex.

7.2 Cache Allocation Technology (CAT)

7.2.1 Overview

Cache Allocation Technology (CAT) allows shared caches to be partitioned at the way level between classes of service. Judicious use of CAT can greatly reduce the jitter in real-time applications due to disturbances in the cache state due to sharing. For optimal real-time performance, the real-time process should have exclusive access to the cache ways it will be using.

The processor provides support for CAT on the Level 2 (L2) cache and last level cache (LLC, also known as L3). Programming interfaces for L2 CAT are architectural and can be discovered by referring to the Intel® 64 and IA-32 Architectures Software Developer's Manual, Chapter 17.19 "Intel® Resource Director Technology (Intel® RDT) Allocation Features." L3 CAT is model specific and non-architectural. As such, programming interfaces for L3 CAT may not align with the Intel Software Developer's Manual and therefore are provided here for completeness.

7.2.2 Architecture

Details on the CAT architecture can be found in *Intel® 64 and IA-32 Architectures Software Developer's Manual,* Chapter 17.19 "Intel® Resource Director Technology (Intel® RDT) Allocation Features."

7.2.3 Usage

Some operating systems and hypervisors provide their own interfaces for configuring CAT. Intel recommends checking with your OS/VMM vendor to determine their support. If your OS/VMM vendor does not support CAT, or you prefer to interface with the MSR's directly, follow the information in the following referenced sections of the Intel Software Developer's Manual and the information in this document.

7.2.3.1 Detecting Cache Allocation Technology Support

To detect the presence of L2 CAT, see Chapter 17.19.4.2 "Cache Allocation Technology: Resource Type and Capability Enumeration" in Volume 3B of the Intel® 64 and IA-32 Architectures Software Developer's Manual.

To detect the presence of L3 CAT, attempt to read MSR 0xC90. If the system generates a fault, L3 CAT is not supported.

7.2.3.2 Detecting Code Data Prioritization (CDP) Support

To determine support for Code Data Prioritization, see Chapter 17.19.4.2 "Cache Allocation Technology: Resource Type and Capability Enumeration" in Volume 3B of the Intel® 64 and IA-32 Architectures Software Developer's Manual.

7.2.3.3 Detecting the Number of Classes of Service

To detect the number of L2 Classes of Service, see Chapter 17.19.4.2 “Cache Allocation Technology: Resource Type and Capability Enumeration” in Volume 3B of the Intel® 64 and IA-32 Architectures Software Developer’s Manual.

The processor has sixteen L3 Classes of Service.

7.2.3.4 Detecting the Capacity Bit Mask Length

To detect the Capacity Bit Mask Length (CBM) for L2 CAT, see Chapter 17.19.4.2 “Cache Allocation Technology: Resource Type and Capability Enumeration” in Volume 3B of the Intel® 64 and IA-32 Architectures Software Developer’s Manual.

The L3 CAT CBMs are defined as follows:

Table 11. L3 CAT CBMs

Processor	CBM
13th Generation Intel® Core™ i9-13900E	36 MiB, 12-ways
13th Generation Intel® Core™ i7-13700E	30 MiB, 10-ways
13th Generation Intel® Core™ i5-13500E	20 MiB, 8-ways
13th Generation Intel® Core™ i5-13400E	20 MiB, 8-ways
13th Generation Intel® Core™ i3-13100E	12 MiB, 12-ways
13th Generation Intel® Core™ i9-13900TE	36 MiB, 12-ways
13th Generation Intel® Core™ i7-13700TE	30 MiB, 10-ways
13th Generation Intel® Core™ i9-13500TE	20MiB, 8-ways
13th Generation Intel® Core™ i9-13100TE	12MIB, 8-ways
13th Generation Intel® Core™ i7-1365URE	12MIB, 4-Ways
13th Generation Intel® Core™ i5-1345URE	12MIB, 4-Ways
13th Generation Intel® Core™ i3-1315URE	10MIB, 4-Ways
13th Generation Intel® Core™ i7-1370PRE	24MIB, 8-Ways
13th Generation Intel® Core™ i5-1350PRE	12MIB, 6-Ways
13th Generation Intel® Core™ i3-1320PRE	12MIB, 6-Ways
13th Generation Intel® Core™ i7-13800HRE	24MIB, 8-Ways
13th Generation Intel® Core™ i5-13600HRE	18MIB, 6-Ways
13th Generation Intel® Core™ i3-13300HRE	12MIB, 6-Ways

7.2.3.5 Intel® Architecture Class of Service Cache Mask Register Details

To detect the L2 CAT cache mask details, see Chapter 17.19.4.3 “Cache Allocation Technology: Cache Mask Configuration” in Volume 3B of the *Intel® 64 and IA-32 Architectures Software Developer’s Manual*.

Since the number of L3 classes of service is sixteen, the L3 CAT cache mask details are as follows:

MSR range: 0xC90 – 0xC9F for COS0 through COS15, respectively.

7.2.3.6 IA Class of Service Register Details

For register details on how to select a class of service, see Chapter 17.19.4.3 “Cache Allocation Technology: Cache Mask Configuration” in Volume 3B of the *Intel® 64 and IA-32 Architectures Software Developer’s Manual*.

There is only one class of service register, referred to as the IA32_PQR_ASSOC, that sets the class of service for both L2 CAT and L3 CAT. This is a per-logical thread.

```
IA32_PQR_ASSOC = MSR 0xC8F
```

Note: RMID is not supported.

7.2.3.7 L3 Class of Service Aliasing

When there is a discrepancy in the number of supported classes of service between the L2 and the L3, as is the case for 11th Generation Intel® Core™ processors, the L3 class of service is aliased using the lower 2 bits of the COS written to IA32_PQR_ASSOC.

Example:

```
Desired COS: 11 (decimal)

IA32_PQR_ASSOC = (0xB << 32) =
1011,0000,0000,0000,0000,0000,0000,0000,0000

       Actual L2 COS = 0xB = 1011’b = 11 (decimal)
       Actual L3 COS = 0x3 = 11’b = 3 (decimal)
```

7.3 Data Direct Input/Output (DDIO) / Write Cache (WRC)

7.3.1 Overview

I/O streams, such as PCIe MMIO read/write, can be considered "temporally local," that is, the data is used by the CPU right after it arrives. Especially in cyclic real-time use cases, placing the data directly in the cache provides lower latency for the CPU when processing the data, and reduces unnecessary memory traffic, improving general performance and power efficiency.

Data Direct Input/Output (DDIO) addresses this issue by allowing I/O devices to have their writes allocate directly into specified ways of the L3 cache. Data Direct IO (DDIO) was originally introduced in Intel® Xeon® processors. A variant of this capability is available on selected 13th Generation Intel® Core™ Processors as the WRC or write cache.

7.3.2 Architecture

Both DDIO and WRC can be considered microarchitectural features. Once enabled in BIOS via Intel® TCC Mode, no additional software involvement is required to use this capability.

7.3.3 Usage

Once enabled in BIOS via the Intel® TCC Mode, no additional effort is needed. However, for optimal results, software should consume the data as quickly as possible to ensure it is consumed from cache rather than memory, as subsequent I/O writes will victimize older data if the CPU has not consumed it.

If Intel® TCC Mode is *enabled*, the following cache ways are allocated to the WRC:
COS3 = 0x001

Note: VCrt is fixed at COS3, so only this capacity bit mask can be selected when WRC is enabled for VCrt.

If Intel® TCC Mode is *disabled*, the following WRC capacity bit mask is set to:
COS3=0x800

7.4 Upstream Virtual Channels

7.4.1 Overview

In real-time applications, it is necessary to enable critical transactions, both reads and writes, to pass low-priority transactions that may be congesting the data path. The PCIe specification from PCI-SIG introduces Virtual Channels (VCs) and the concept of Traffic Classes (TCs), which enable PCIe transactions to be prioritized by applying service policies.

There is still a need to optimize the entire data path. The concept of fabric VCs address this by extending the PCIe Virtual Channel concept to the internal data buses.

The fabric provides multiple VCs that differentiate between traffic types by distributing and ordering traffic between VCs, prioritizing VCs through arbitration, and tailoring VC attributes to each VC's specific purpose.

Note: No ordering relationship is enforced between transactions on different VCs. Thus, transactions within one VC may pass those of another within the fabric.

VCs are given a “VCx” designation where “x" is some identifier (for example, VC0, VC1, VC1a, VC1b, VCrt, VCbr). A general-purpose VC (often designated VC0) optimized for high bandwidth and given standard priority is the default VC for generic traffic.

VCs are implemented on a per-hop basis, so each IP and each subcomponent within an IP must explicitly enable each VC. Some IPs possibly do not support one or more VCs, thus creating a bottleneck where VC transactions from multiple VCs share queues. This bottleneck eliminates differentiation and serializes the data path. To take full advantage of this feature, end-to-end (E2E) VCs support VCs along each hop between source and destination of a given data path.

To support real-time applications, the fabric implements a low-bandwidth, high-priority VC that is generically and sometimes architecturally referred to as VCrt. In this context, “Virtual Channels” henceforth refers to support for VCrt.

7.4.2 Architecture

7.4.2.1 PCIe Virtual Channels

PCIe Virtual Channels are limited in scope to the PCIe controllers, switches, and devices that make up the local PCIe hierarchy of a platform. While in practice, the same TC and VC concepts are usually shared end to end, the fabric may implement VCs differently than PCIe while still adhering to the PCI-SIG specification.

The CPU attached PCIe controllers do not see significant increase in latency from high bandwidth congestion, while the PCH attached PCIe controller does, so multi-VC is supported to alleviate this issue.

A PCIe root port that supports PCIe VCs must correctly configure the VC Capability structure to denote which VCs are supported, and the TC/VC mapping. The VC Capability structure is defined in the PCIe Base Specification.

7.4.2.1.1 PCIe Controller Virtual Channel Support

Table 12. PCIe Controller Virtual Channel Support

PCIe Controller	VC Support
CPU Attached Gen5	VC0 only
CPU Attached Gen4	VC0 and VC1
PCH Attached Gen3	VC0 only

7.4.2.1.2 PCIe Virtual Channel Mapping Capabilities

Table 13. PCIe Virtual Channel Mapping Capabilities

TC	VC Mapping Capability
TC0	VC0 best-effort
TC1-TC3	VC1 real-time

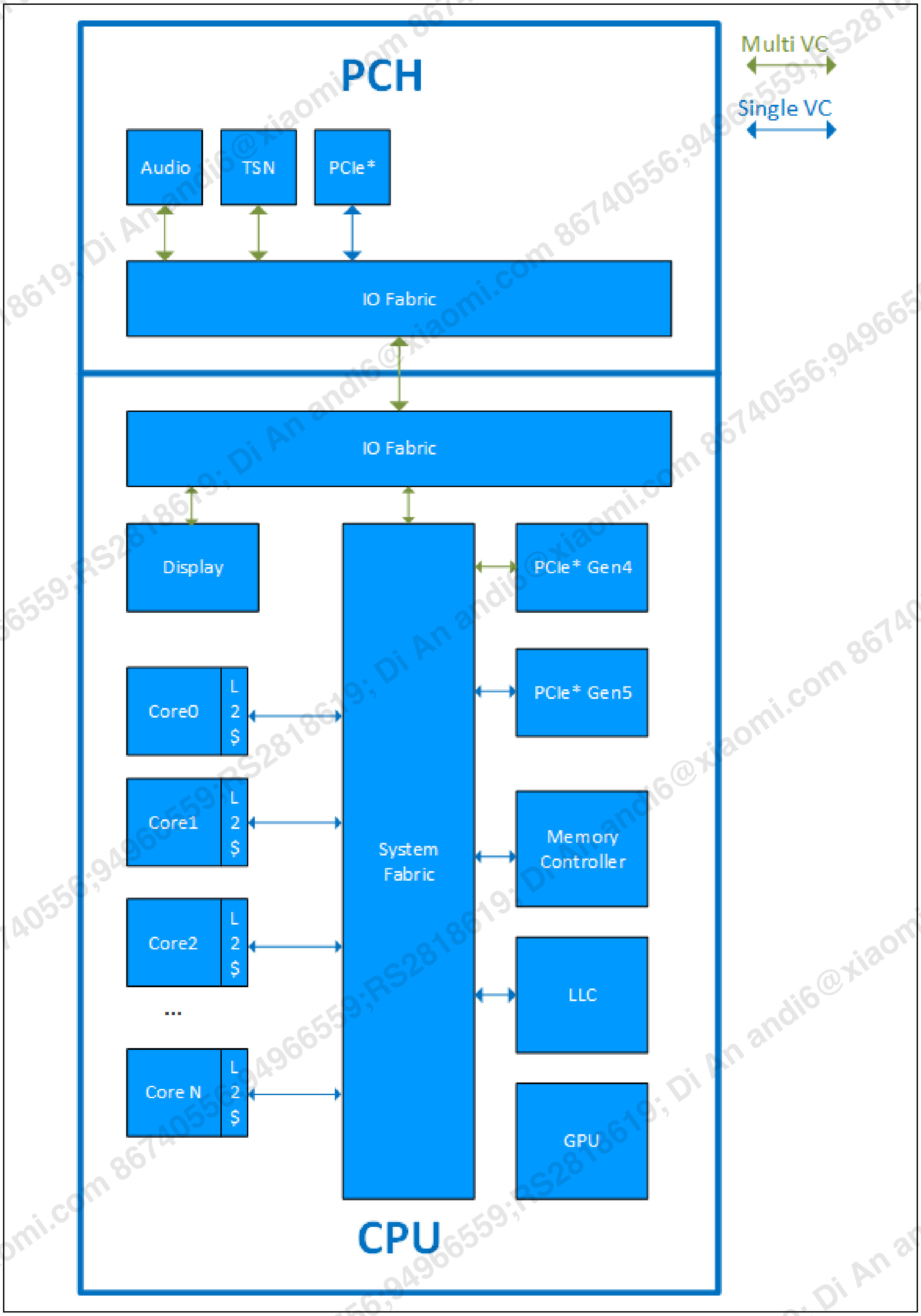
7.4.2.2 Fabric Virtual Channels

Fabric Virtual Channels include VC implementation in the SoC fabric from the coherent domain through the I/O fabric. Each stage in the hardware pipeline from the source of a transaction (for example, PCIe) to the destination (for example, memory) may implement VCs to the next hop upstream or downstream. Any stage that doesn't implement VCs may cause a bottleneck.

The main IPs that implement fabric VCs include:

- IOSF (I/O Scalable Fabric)
- CMF (Coherent Memory Fabric)

Figure 4. Fabric Virtual Channels Block Diagram



7.4.3 Usage

With Intel® TCC Mode enabled, TC0 uses the high-bandwidth, best-effort virtual channel, and TC1–TC3 use the low-bandwidth, real-time virtual channel.

For register specifications, see the BIOS Writer's Guide listed in Reference Documents.

7.5 Time Synchronization

Time synchronization features are available by default; no platform tuning is needed to enable these features.

Intel® provides sample applications that demonstrate use of these features.

§

8.0 Fabric Tuning

Fabric tuning includes microarchitecture-specific optimizations for functional components in the fabric that include prioritization, arbitration, and flow control. These tunings come at a high cost in implementation time and performance tradeoffs, and they provide comparatively little improvement relative to the other types of tuning. As such, manual fabric tuning is not recommended.

However, since the best possible real-time performance can be attained by tuning the fabric arbitration and prioritization characteristics, Intel will provide guidance on fabric tuning for 13th Generation Intel® Core® processors upon demand. Contact your Intel representative for more information.

§

9.0 Terminology

Table 14. 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 Length
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.
Intel® DDIO	Intel® Data Direct Input/ Output
DE	Display Engine
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

Term	Description
Intel® TCC	Intel® Time Coordinated Computing A modern approach to architecting distributed, synchronized, scalable computing systems that address real-time application requirements for cyber-physical systems (CPS). Intel® TCC moves beyond traditional real-time systems based on simple microcontrollers.
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

Term	Description
P-State	A power-performance, implantation-dependent state of devices or processors that indicate power and frequency levels.
PCH	Platform Controller Hub
PCS	Physical Coding Sublayer
PCIe*	Peripheral Component Interconnect express*
PM	Power Management
PTM	Precision Time Measurement
PREEMPT_RT	The PREEMPT_RT patch (also the -rt patch or RT patch) makes Linux* into a Real-Time Operating System (RTOS) to a large degree.
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
RTOS	Real-Time Operating System
SA	System Agent
SGMII	Serial Gigabit Media-Independent Interface
SKU	Stock Keeping Unit
STA	Station Management
TBI	Ten-Bit Interface
TCs	Traffic Classes
TDP	Thermal Design Power

Term	Description
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. Intel® 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, computes 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.
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.

§

10.0 Reference Documents

In Table 15, 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 15. Reference Documents

Document	Document No./Location
Real-Time Gold Deck	627170 https://cdrdv2.intel.com/v1/dl/getContent/627170
13th Generation Intel® Core™ Mobile Processors for IoT Edge Page	https://www.intel.com/content/www/us/en/products/details/embedded-processors/core/13thgenmobile.html
13th Generation Intel® Core™ Desktop Processors for IoT Edge Page	https://www.intel.com/content/www/us/en/products/details/embedded-processors/core/13thgen.html
Raptor Lake Processor External Design Specification	640555 https://cdrdv2.intel.com/v1/dl/getContent/640555
Raptor Lake-P for IoT Platforms External Design Specification (EDS) Addendum	735978 https://cdrdv2.intel.com/v1/dl/getContent/735978
Raptor Lake-S for IoT Platforms External Design Specification (EDS) Addendum	709998 https://cdrdv2.intel.com/v1/dl/getContent/709998
Raptor Lake -S for IoT Platforms BIOS Writer's Guide - Addendum	732579 https://cdrdv2.intel.com/v1/dl/getContent/732579

Document	Document No./Location
Yocto Project*-based Board Support Package for Raptor Lake -S on IoT Platforms - Getting Started Guide (Beta Release)	730934 https://cdrdv2.intel.com/v1/dl/getContent/730934
Yocto Project*-based Board Support Package for Raptor Lake -S for IoT Platforms - Release Notes (Beta Release)	730933 https://cdrdv2.intel.com/v1/dl/getContent/730933
Raptor Lake -P Yocto-Project* BSP Release Note	736744 https://www.intel.com/content/www/us/en/secure/content-details/736744
Ubuntu with Kernel Overlay on Raptor Lake - P for Edge Platforms - Getting Started Guide	762654 https://cdrdv2.intel.com/v1/dl/getContent/762654
Ubuntu with Kernel Overlay for Raptor Lake – P for Edge Platforms (Release Note)	762655 https://cdrdv2.intel.com/v1/dl/getContent/762655
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
Intel Ethernet Controller I225/I226 Product Brief	https://cdrdv2.intel.com/v1/dl/getContent/621753
Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3B: System programming guide, part 2	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://docs.yoctoproject.org/kernel-dev/index.html

Document	Document No./Location
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
Intel® 64 and IA-32 Architectures Software Developer's Manual	https://software.intel.com/en-us/articles/intel-sdm
PCI-SIG, PCI Express Base Specification Revision 2.1 (subscription based)	http://pcisig.com/specifications/pciexpress/base
Linux kernel command-line parameters	• https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/Documentation/admin-guide/kernel-parameters.txt?h=v5.15 • https://www.linuxtopia.org/online_books/linux_kernel/kernel_configuration/re46.html
PREEMPT_RT patches	https://wiki.linuxfoundation.org/realtime/start

§

Appendix A Variant Kernel Configurations

This appendix details the kernel boot parameters for tuning the performance of the platform. The parameters are based on kernel version 5.10 with the PREEMPT_RT patch and tested on BIOS TGLIFUI1.R00.3313.A03.2008071806.

Table 16. Kernel Command-Line Parameters for RTCP/XDP and Cyclic Test

Kernel Command-Line Parameter	RTCP Value	Cyclic Test
cat	✓	✓
/proc/cmdline	✓	✓
processor.max_cstate=0	✓	✓
intel.max_cstate=0	✓	✓
processor_idle.max_cstate=0	✓	✓
intel_idle.max_cstate=0	✓	✓
clocksource=tsc	✓	✓
tsc=reliable	✓	✓
nowatchdog	No value required	No value required
nmi_watchdog=0	✓	✓
intel_pstate=disable	✓	✓
nosoftlockup	✓ ***Note:*** No value required	✓ ***Note:*** No value required
idle=poll	✓	✓
noht	✓	✓
isolcpus=2-3	✓	✓
rcu_nocbs=2-3	✓	✓
irqaffinity=0	✓	✓
rcupdate.rcu_cpu_stall_suppress=1	✓	✓
rcu_nocb_poll	✓	✓
i915.enable_rc6=0	✓	✓
i915.enable_dc=0	✓ ***Note:*** Option is only required when the platform includes integrated graphics.	✓ ***Note:*** Option is only required when the platform includes integrated graphics.

Kernel Command-Line Parameter	RTCP Value	Cyclic Test
i915.disable_power_well=0	✓ ***Note:*** Option is only required when the platform includes integrated graphics.	✓ ***Note:*** Option is only required when the platform includes integrated graphics.
hugepages=1024	✓	✓
mce=off	✓	✓
hpet=disable	✓	✓
numa_balancing=disable	✓	✓
igb.blacklist	✓	✓
efi=runtime	✓	✓
BOOT_IMAGE=(hd0,gpt2)/boot/bzImage-linux-intel-ese-lts-rt-5.10-kernel	✓	✓
root=PARTLABEL=primary	✓	✓
apparmor=1	✓	✓
security=apparmor	✓	✓
mem_sleep_default=deep	✓	✓
mender.efi=PARTLABEL=efi	✓	✓
mender.primary=PARTLABEL=primary	✓	✓
mender.secondary=PARTLABEL=secondary	✓	✓
mender.data=PARTLABEL=data	✓	✓

mender.swap=PARTLABEL=swap
resume=PARTLABEL=swap
	✓	✓

rootwait	✓ ***Note:*** No value required	✓
console=ttyS0,115200	✓	✓
console=tty0	✓	✓
art=virtallow	✓	✓
iommu=pt	✓	✓
nohz_full=2-3	✓	---
init=/sbin/preinit-env	✓	✓
console=ttyS4,115200n8	✓	✓
console=ttyS5,115200n8	✓	✓
i915.force_probe=*	✓	✓
i915.enable_guc=2	✓	✓
udmabuf.list_limit=8192	✓	✓
i915.enable_guc=7	✓	✓

§