AMD EPYC

TUNING GUIDE
AMD EPYC 9004

BIOS & Workload

Publication	58011
Revision	1.3
Issue Date	June, 2023

Date	Version	Changes
July, 2022	0.1	Initial NDA partner release
Sep, 2022	0.2	Updated BIOS naming and defaults
Nov, 2022	1.0	Initial public release
Dec, 2022	1.1	Minor errata corrections
Mar, 2023	1.2	Added 97xx OPN and AMD 3D V-Cache™ technology information
Jun, 2023	1.3	Second public release

Audience

This tuning guide describes best practices for optimizing performance using AMD default BIOS settings. It is intended for a technical audience such as system architects, production deployment, and performance engineering teams with:

- A background in configuring servers.
- Administrator-level access to both the server management Interface (BMC) and the OS.
- Familiarity with both the BMC and OS-specific configuration, monitoring, and troubleshooting tools.

Authors

Muhammad Ashfaq, Anil Rajput, and Jesse Rangel

Note: All of the settings described in this Tuning Guide apply to all AMD EPYC 9004 Series Processors of all core counts with or without AMD 3D V-Cache™ except where explicitly noted otherwise.

Table of Contents

This page intentionally left blank.

Chapter 1 Introduction

Default BIOS options generally produce the best overall performance for generic workloads, but these defaults may not be optimal for a specific workload. AMD continually tests various workloads; this tuning guide discusses BIOS options to deliver both maximum performance and performance-per- watt (power efficiency).

- "BIOS Defaults Summary" on page 13 provides a quick overview of default AMD EPYC 9004 BIOS settings.
- "BIOS Option Details" on page 23 provides detailed information about the AMD EPYC 9004 BIOS options and the potential benefit of modifying each one.
- "Workload-Specific BIOS Settings" on page 35 presents sample workloads and recommended BIOS settings. Keep in mind that these BIOS settings are not "one size fits all" because your specific workload(s) are not identical to synthetic benchmarks.

Note: Not all platforms support all of the BIOS settings described in this Tuning Guide. Please contact your platform vendor if you cannot see one or more needed settings.

AMD strongly recommends that customers download and install the latest BIOS update for your AMD EPYC 9004 Series Processor-based server from your platform vendor. BIOS updates often help customers by providing new and updated features, bug fixes, enhancements, security features, and other improvements. These improvements can help your system software stability and dependency modules (such as hardware, firmware, drivers, and software) by giving you a more robust environment to run your applications.

This page intentionally left blank.

Chapter 2

AMD EPYC™ 9004 Series Processors

AMD EPYC™ 9004 Series Processors represent the fourth generation of AMD EPYC server-class processors. This generation of AMD EPYC processors feature AMD's latest "Zen 4" based compute cores, next-generation Infinity Fabric, next-generation memory & I/O technology, and use the new SP5 socket/packaging.

2.1 General Specifications

AMD EPYC 9004 Series Processors offer a variety of configurations with varying numbers of cores, Thermal Design Points (TDPs), frequencies, cache sizes, etc. that complement AMD's existing server portfolio with further improvements to performance, power efficiency, and value. Table 1-1 lists the features common to all AMD EPYC 9004 Series Processors.

Common Features of all AMD EPYC 9004 Series Processors	
Compute cores	Zen4-based
Core process technology	5nm
Maximum cores per Core Complex (CCX)	8
Max memory per socket	6 TB
Max # of memory channels	12 DDR5
Max memory speed	4800 MT/s DDR5
Max lanes Compute eXpress Links	64 lanes CXL 1.1+
Max lanes Peripheral Component Interconnect	128 lanes PCIe® Gen 5

Table 2-1: Common features of all AMD EPYC 9004 Series Processors

2.2 Model-Specific Features

Different models of 4th Gen AMD EPYC processors have different feature sets, as shown in Table 1-2.

AMD EPYC 9004 Series Processor (Family 19h) Features by Model		
Codename	"Genoa"*	"Bergamo"*
Model #	91xx-96xx	97xx
Max number of Core Complex Dies (CCDs)	12	8
Number of Core Complexes (CCXs) per CCD	1	2
Max number of cores (threads)	96 (192)	128 (256)
Max L3 cache size (per CCX)	1,152 MB (96 MB)♦	256 MB (16 MB)
Max Processor Frequency	4.4 GHz♦♦	3.15 GHz
Includes ♦AMD 3D V-Cache (9xx4X) and ♦♦high-frequency (9xx4F) models.		
*GD-122: The information contained herein is for informational purposes only and is subject to change without notice. Timelines, roadmaps, and/or product release dates shown herein and plans only and subject to change. "Genoa" and "Bergamo" are codenames for AMD architectures and are not product names.		

Table 2-2: AMD EPYC 9004 Series Processors features by model

2.3 Operating Systems

AMD recommends using the latest available targeted OS version and updates. Please see AMD EPYC™ Processors Minimum Operating System (OS) Versions for detailed OS version information.

2.4 Processor Layout

AMD EPYC 9004 Series Processors incorporate compute cores, memory controllers, I/O controllers, RAS (Reliability, Availability, and Serviceability), and security features into an integrated System on a Chip (SoC). The AMD EPYC 9004 Series Processor retains the proven Multi-Chip Module (MCM) Chiplet architecture of prior successful AMD EPYC processors while making further improvements to the SoC components.

The SoC includes the Core Complex Dies (CCDs), which contain Core Complexes (CCXs), which contain the "Zen 4"-based cores. The CCDs surround the central high-speed I/O Die (and interconnect via the Infinity Fabric). The following sections describe each of these components.

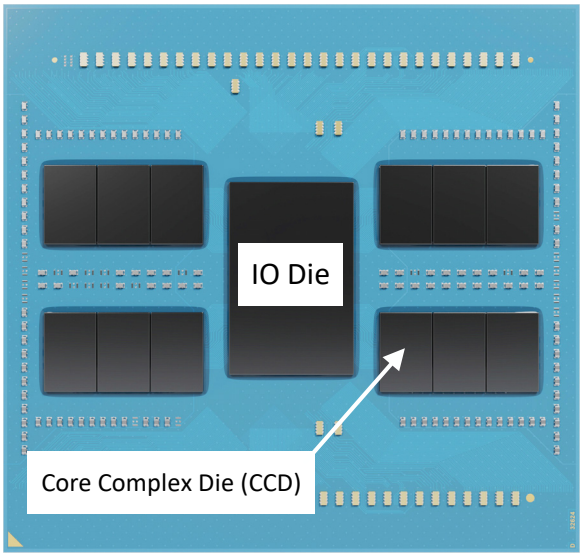

Figure 2-1: AMD EPYC 9004 configuration with 12 Core Complex Dies (CCD) surrounding a central I/O Die (IOD)

2.5 "Zen 4" Core

AMD EPYC 9004 Series Processors are based on the new "Zen 4" compute core. The "Zen 4" core is manufactured using a 5nm process and is designed to provide an Instructions per Cycle (IPC) uplift and frequency improvements over prior generation "Zen" cores. Each core has a larger L2 cache and improved cache effectiveness over the prior generation. Each "Zen 4" core includes:

- Up to 32 KB of 8-way L1 I-cache and 32 KB of 8-way of L1 D-cache
- Up to a 1 MB private unified (Instruction/Data) L2 cache.

Each core supports Simultaneous Multithreading (SMT), which allows 2 separate hardware threads to run independently, sharing the corresponding core's L2 cache.

2.6 Core Complex (CCX)

Figure 2-2 shows a Core Complex (CCX) where up to eight "Zen 4"-based cores share a L3 or Last Level Cache (LLC). Enabling Simultaneous Multithreading (SMT) allows a single CCX to support up to 16 concurrent hardware threads.

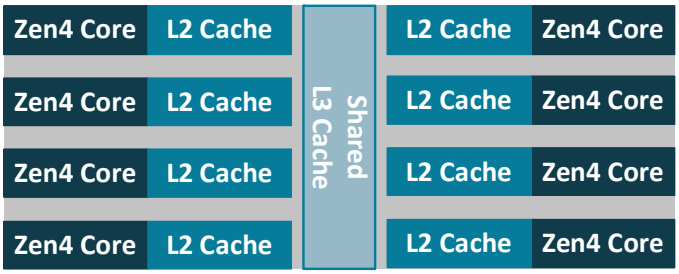

Figure 2-2: Top view of 8 compute cores sharing an L3 cache (91xx-96xx models)

2.7 Core Complex Dies (CCDs)

The Core Complex Die (CCD) in an AMD EPYC 9xx4 Series Processor may contain either one or two CCXs, depending on the processor (91xx-96xx "Genoa" vs. 97xx "Bergamo"), as shown in Figure 2-5.

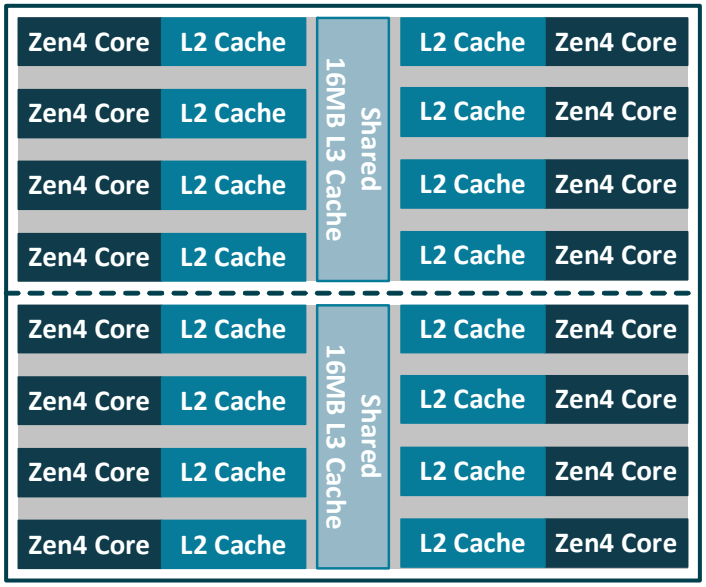

Figure 2-3: 2 CCXs in a single 4th Gen AMD EPYC 97xx CCD

Each of the Core Complex Dies (CCDs) in a 97xx model AMD EPYC 9004 Series Processor contains two CCXs (Figure 2-5):

AMD EPYC 9004 Series Processor	91xx-96xx	97xx
# of CCXs within a CCD	1	2

Table 2-3: CCXs per CCD by AMD EPYC model

You can disable cores in BIOS using one or both of the following approaches:

- Reduce the cores per L3 from 8 down to 7,6,5,4,3,2, or 1 while keeping the number of CCDs constant. This approach increases the effective cache per core ratio but reduces the number of cores sharing the cache.
- Reduce the number of active CCDs while keeping the cores per CCD constant. This approach maintains the advantages of cache sharing between the cores while maintaining the same cache per core ratio.

2.8 AMD 3D V-Cache™ Technology

AMD EPYC 9xx4X Series Processors include AMD 3D V-Cache™ die stacking technology that enables 97xx to achieve more efficient chiplet integration. AMD 3D Chiplet architecture stacks L3 cache tiles vertically to provide up to 96MB of L3 cache per die (and up to 1 GB L3 Cache per socket) while still providing socket compatibility with all AMD EPYC™ 9004 Series Processor models.

AMD EPYC 9004 Series Processors with AMD 3D V-Cache technology employ industry-leading logic stacking based on copper-to-copper hybrid bonding "bumpless" chip-on-wafer process to enable over 200X the interconnect densities of current 2D technologies (and over 15X the interconnect densities of other 3D technologies using solder bumps), which translates to lower latency, higher bandwidth, and greater power and thermal efficiencies.

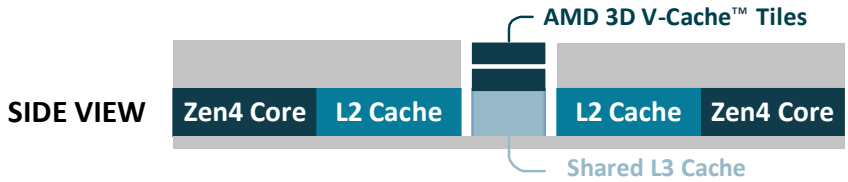

Figure 2-4: Side view of vertically-stacked central L3 SRAM tiles

AMD EPYC 9004 Series Processors	9xx4	9004X (with 3D V-Cache)
Max Shared L3 Cache per CCD	32 MB	96 MB

Table 2-4: L3 cache by processor model

Different OPNs also may have different numbers of cores within the CCX. However, for any given part, all CCXs will always contain the same number of cores.

2.9 I/O Die (Infinity Fabric™)

The CCDs connect to memory, I/O, and each other through an updated I/O Die (IOD). This central AMD Infinity Fabric™ provides the data path and control support to interconnect CCXs, memory, and I/O. Each CCD connects to the IOD via a dedicated high-speed Global Memory Interconnect (GMI) link. The IOD helps maintain cache coherency and additionally provides the interface to extend the data fabric to a potential second processor via its xGMI, or G-links. AMD EPYC 9004 Series Processors support up to 4 xGMI (or G-links) with speeds up to 32Gbps. The IOD exposes DDR5 memory channels, PCIe® Gen5, CXL 1.1+, and Infinity Fabric links.

All dies (chiplets) interconnect with each other via AMD Infinity Fabric technology. Figure 2-6 (which corresponds to Figure 2-2, above) shows the layout of a 96-core AMD EPYC 9654 processor. The AMD EPYC 9654 has 12 CCDs, with each CCD connecting to the IOD via its own GMI connection.

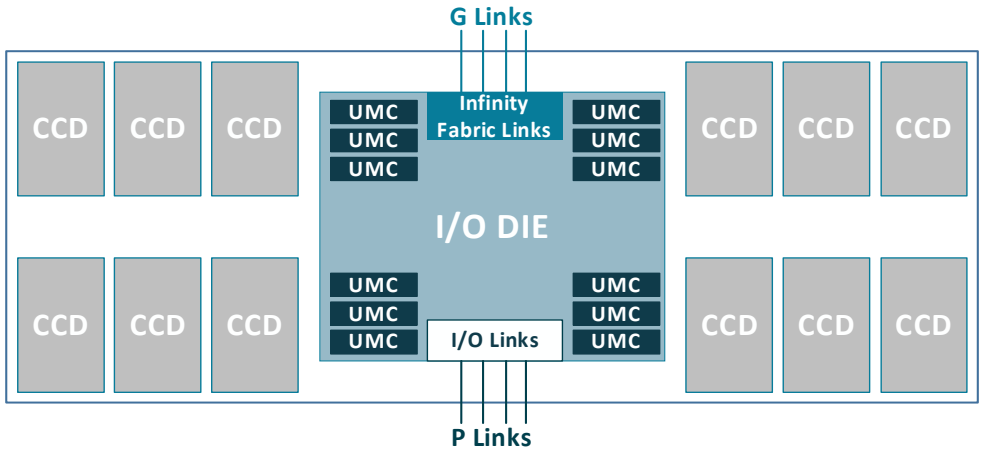

Figure 2-5: AMD EPYC 9654 processor internals interconnect via AMD Infinity Fabric (12 CCD processor shown)

AMD also provides "wide" OPNs (e.g. AMD EPYC 9334) where each CCD connects to two GMI3 interfaces, thereby allowing double the Core-to-I/O die bandwidth.

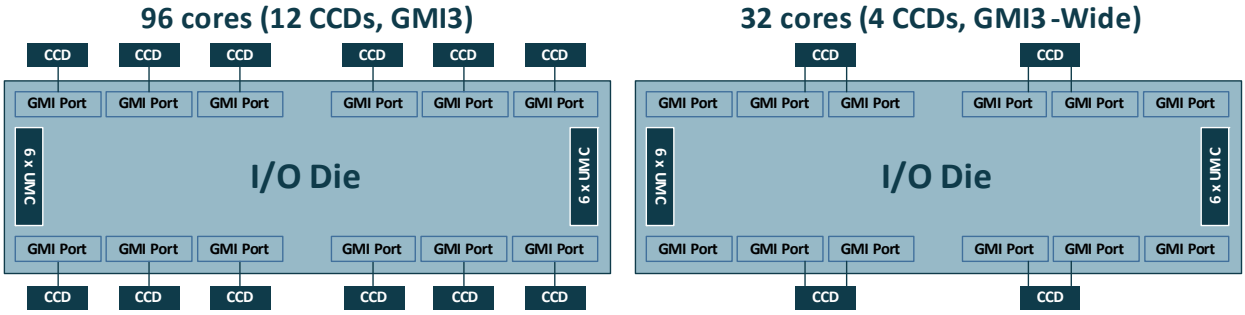

Figure 2-6: Standard vs. Wide GMI links

The IOD provides twelve Unified Memory Controllers (UMCs) that support DDR5 memory. The IOD also presents 4 'P-links' that the system OEM/designer can configure to support various I/O interfaces, such as PCIe Gen5, and/or CXL 1.1+.

2.10 Memory and I/O

Each UMC can support up to 2 DIMMs per channel (DPC) for a maximum of 24 DIMMs per socket. OEM server configurations may allow either 1 DIMM per channel or 2 DIMMs per channel. 4th Gen AMD EPYC processors can support up to 6TB of DDR5 memory per socket. Having additional and faster memory channels compared to previous generations of AMD EPYC processors provides additional memory bandwidth to feed high-core-count processors. Memory interleaving on 2, 4, 6, 8, 10, and 12 channels helps optimize for a variety of workloads and memory configurations.

Each processor may have a set of 4 P-links and 4 G-links. An OEM motherboard design can use a G-link to either connect to a second 4th Gen AMD EPYC processor or to provide additional PCIe Gen5 lanes. 4th Gen AMD EPYC processors support up to eight sets of x16-bit I/O lanes, that is, 128 lanes of high-speed PCIe Gen5 in single-socket platforms and up to 160 lanes in dual-socket platforms. Further, OEMs may either configure 32 of these 128 lanes as SATA lanes and/or configure 64 lanes as CXL 1.1+. In summary, these links can support:

- Up to 4 G-links of AMD Infinity Fabric connectivity for 2P designs.
- Up to 8 x16 bit or 128 lanes of PCIe Gen 5 connectivity to peripherals in 1P designs (and up to 160 lanes in 2-socket designs).
- Up to 64 lanes (4 P-links) that can be dedicated to Compute Express Link (CXL) 1.1+ connectivity to extended memory.
- Up to 32 I/O lanes that can be configured as SATA disk controllers.

2.11 Visualizing AMD EPYC 9004 Series Processors (Family 19h)

This section depicts AMD EPYC 9004 Series Processors that have been set up with four nodes per socket (NPS=4). Please see "NUMA Topology" on page 10 for more information about nodes.

2.11.1 Models 91xx-96xx ("Genoa")

4th Gen AMD EPYC 9004 processors with model numbers 91xx-96xx have up to 12 CCDs that each contain a single CCX, as shown below.

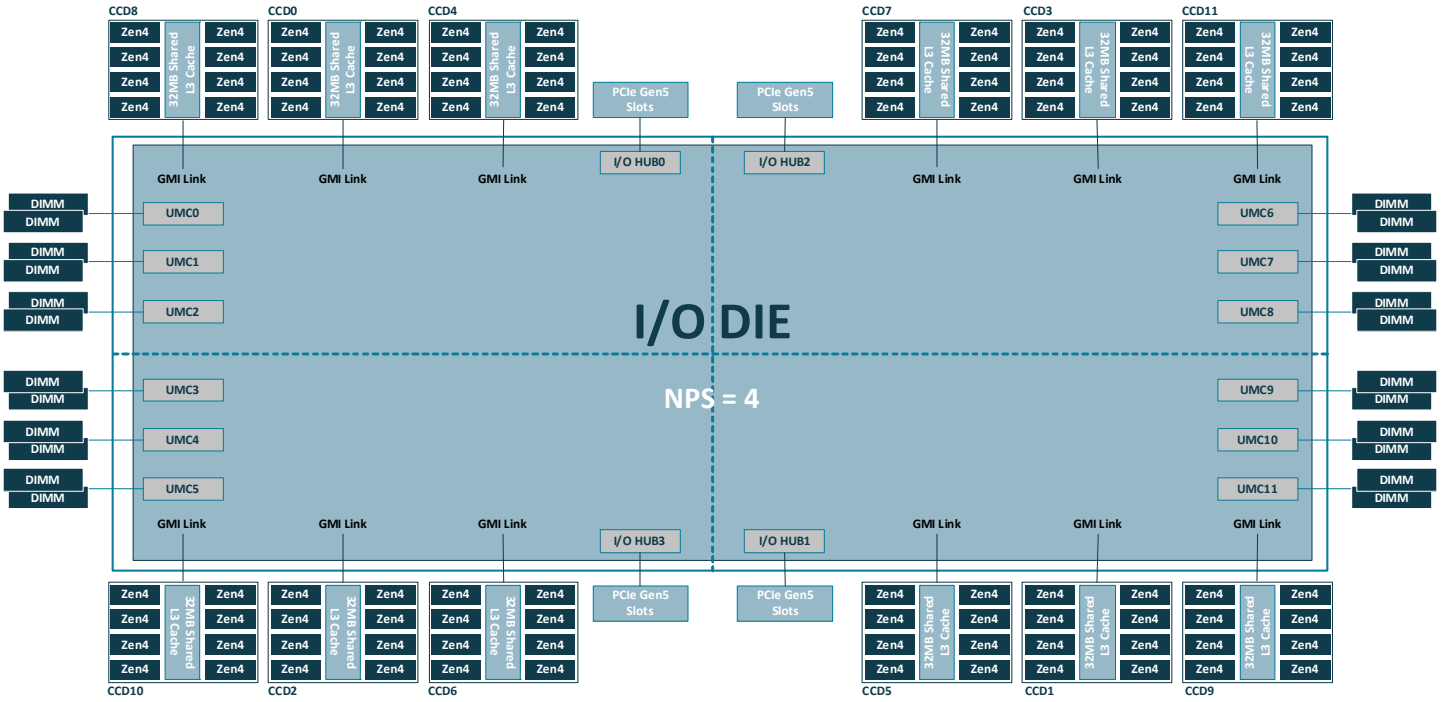

Figure 2-7: The AMD EPYC 9004 SoC consists of up to 12 CCDs and a central IOD for 91xx-96xx models, including "X" OPNs

2.11.2 Models 97xx ("Bergamo")

97xx 4th Gen AMD EPYC 9004 Series Processors with model numbers 97xx have up to 8 CCDs that each contain two CCXs, as shown below.

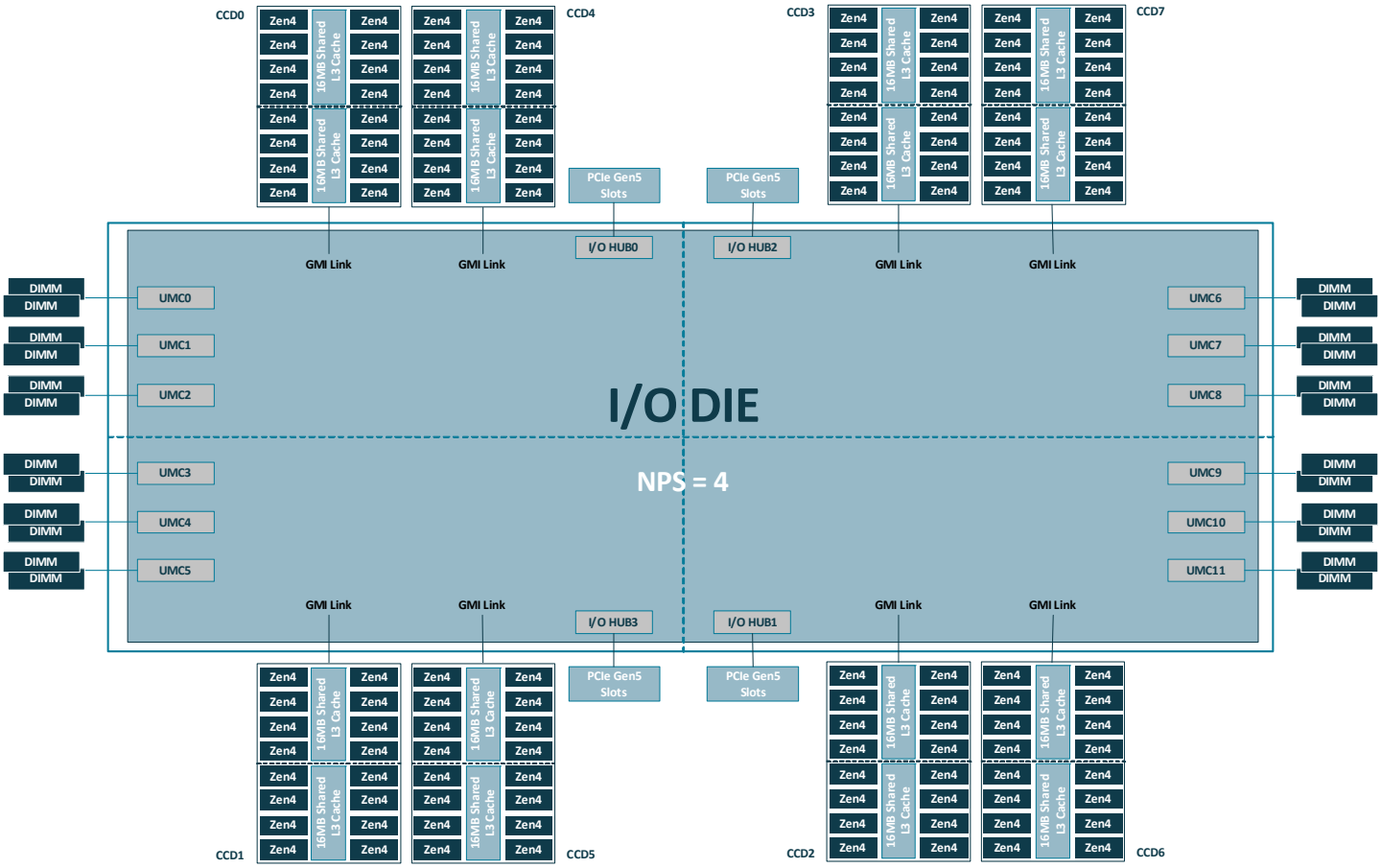

Figure 2-8: The AMD EPYC 9004 System on Chip (SoC) consists of up to 8 CCDs and a central IOD for 97xx models

2.12 NUMA Topology

AMD EPYC 9004 Series Processors use a Non-Uniform Memory Access (NUMA) architecture where different latencies may exist depending on the proximity of a processor core to memory and I/O controllers. Using resources within the same NUMA node provides uniform good performance, while using resources in differing nodes increases latencies.

2.12.1 NUMA Settings

A user can adjust the system **NUMA Nodes Per Socket** (NPS) BIOS setting to optimize this NUMA topology for their specific operating environment and workload. For example, setting NPS=4 as shown in "Memory and I/O" on page 8 divides the processor into quadrants, where each quadrant has 3 CCDs, 3 UMCs, and 1 I/O Hub. The closest processor-memory I/O distance is between the cores, memory, and I/O peripherals within the same quadrant. The furthest distance is between a core and memory controller or IO hub in cross- diagonal quadrants (or the other processor in a 2P configuration). The locality of cores, memory, and IO hub/devices in a NUMA-based system is an important factor when tuning for performance.

The NPS setting also controls the interleave pattern of the memory channels within the NUMA Node. Each memory channel within a given NUMA node is interleaved. The number of channels interleaved decreases as the NPS setting gets more granular. For example:

- A setting of NPS=4 partitions the processor into four NUMA nodes per socket with each logical quadrant configured as its own NUMA domain. Memory is interleaved across the memory channels associated with each quadrant. PCIe devices will be local to one of the four processor NUMA domains, depending on the IOD quadrant that has the corresponding PCIe root complex for that device.

- A setting of NPS=2 configures each processor into two NUMA domains that groups half of the cores and half of the memory channels into one NUMA domain, and the remaining cores and memory channels into a second NUMA domain. Memory is interleaved across the six memory channels in each NUMA domain. PCIe devices will be local to one of the two NUMA nodes depending on the half that has the PCIe root complex for that device.

- A setting of NPS=1 indicates a single NUMA node per socket. This setting configures all memory channels on the processor into a single NUMA node. All processor cores, all attached memory, and all PCIe devices connected to the SoC are in that one NUMA node. Memory is interleaved across all memory channels on the processor into a single address space.

- A setting of NPS=0 indicates a single NUMA domain of the entire system (across both sockets in a two-socket configuration). This setting configures all memory channels on the system into a single NUMA node. Memory is interleaved across all memory channels on the system into a single address space. All processor cores across all sockets, all attached memory, and all PCIe devices connected to either processor are in that single NUMA domain.

You may also be able to further improve the performance of certain environments by using the **LLC (L3 Cache) as NUMA** BIOS setting to associate workloads to compute cores that all share a single LLC. Enabling this setting equates each shared L3 or CCX to a separate NUMA node, as a unique L3 cache per CCD. A single AMD EPYC 9004 Series Processor with 12 CCDs can have up to 12 NUMA nodes when this setting is enabled.

Thus, a single EPYC 9004 Series Processor may support a variety of NUMA configurations ranging from one to twelve NUMA nodes per socket.

Note: If software needs to understand NUMA topology or core enumeration, it is imperative to use documented Operating System (OS) APIs, well-defined interfaces, and commands. Do not rely on past assumptions about settings such as APICID or CCX ordering.

2.13 Dual-Socket Configurations

AMD EPYC 9004 Series Processors support single- or dual-socket system configurations. Processors with a 'P' suffix in their name are optimized for single-socket configurations (see the "Processor Identification" chapter) only. Dual-socket configurations require both processors to be identical. You cannot use two different processor Ordering Part Numbers (OPNs) in a single dual-socket system.

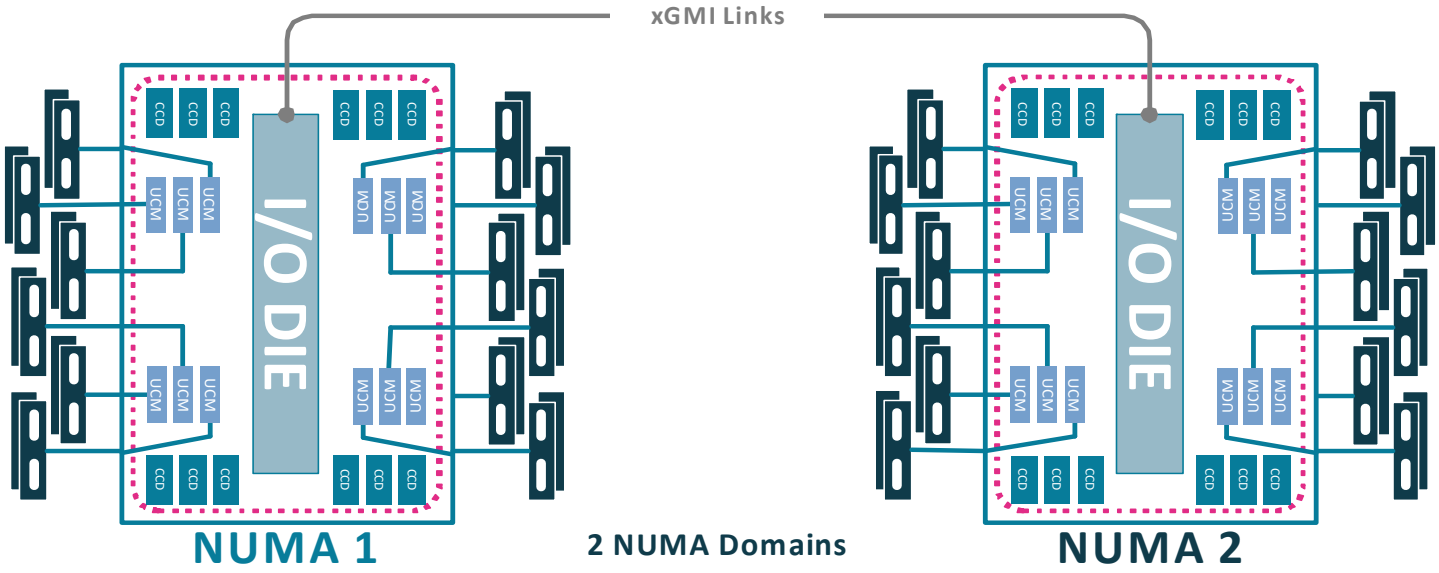

Figure 2-9: Two EPYC 9004 Processors connect through 4 xGMI links (NPS1)

In dual-socket systems, two identical EPYC 9004 series SoCs are connected via their corresponding External Global Memory Interconnect [xGMI] links. This creates a high bandwidth, low latency interconnect between the two processors. System manufacturers can elect to use either 3 or 4 of these Infinity Fabric links depending upon I/O and bandwidth system design objectives.

The Infinity Fabric links utilize the same physical connections as the PCIe lanes on the system. Each link uses up to 16 PCIe lanes. A typical dual socket system will reconfigure 64 PCIe lanes (4 links) from each socket for Infinity Fabric connections. This leaves each socket with 64 remaining PCIe lanes, meaning that the system has a total of 128 PCIe lanes. In some cases, a system designer may want to expose more PCIe lanes for the system by reducing the number of Infinity Fabric G-Links from 4 to 3. In these cases, the designer may allocate up to 160 lanes for PCIe (80 per socket) by utilizing only 48 lanes per socket for Infinity Fabric links instead of 64.

A dual-socket system has a total of 24 memory channels, or 12 per socket. Different OPNs can be configured to support a variety of NUMA domains.

Chapter 3

BIOS Defaults Summary

This chapter provides high-level lists of the default AMD EPYC 9004 BIOS settings and their default values. Please see Chapter 4 of the *BIOS & Workload Tuning Guide for AMD EPYC™ 9004 Series Processors* (available from AMD EPYC Tuning Guides) for detailed descriptions. Later chapters in this Tuning Guide discuss the BIOS options as they relate to a specific workload or set of workloads.

Note: The default setting names and values described in this chapter are the AMD default names and values that serve as recommendations for OEMs. End users must confirm their OEM BIOS setting availability and options.

AMD strongly recommends that customers download and install the latest BIOS update for your AMD EPYC 9004 Series Processor-based server from your platform vendor. BIOS updates often help customers by providing new and updated features, bug fixes, enhancements, security features, and other improvements. These improvements can help your system software stability and dependency modules (such as hardware, firmware, drivers, and software) by giving you a more robust environment to run your applications.

3.1 Processor Core Settings

Name	Default	Description
SMT Control	Auto	• **Enabled/Auto:** Two hardware threads per core. • **Disabled:** Single hardware thread per core.
L1 Stream HW Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L1 Stride Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L1 Region Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L1 Burst Prefetch Mode	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L2 Stream HW Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L2 Up/Down Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
Core Performance Boost	Auto	• **Enabled/Auto:** Enables Core Performance Boost. • **Disabled:** Disables Core Performance Boost.
BoostFmaxEn	Auto	• **Auto:** Use the default Fmax • **Manual:** User can set the boost Fmax
BoostFmax	Auto	Specify the boost Fmax frequency limit to apply to all cores (MHz in decimal)
Global C-State Control	Auto	• **Enabled/Auto:** Controls IO based C-state generation and DF C-states, including core processor C-States • **Disabled:** AMD strongly recommends not disabling this option because this also disables core processor C-States.

Table 3-1: Processor core BIOS settings

X3D	Auto	Enables or disables AMD 3D V-Cache™ technology on Cache Optimized (9004X) processors. • **Auto:** Enabled on an AMD EPYC 9004 Series processor with AMD 3D V-Cache™ technology, enabling this option enables the AMD 3D V-Cache module in the CCD to increase the total size of the L3 cache memory to 96MB • **Disabled:** Disabling this option reduces the L3 cache in the CCD to 32MB. *Note: This option only applies to AMD EPYC 9004 Series Processors with AMD 3D V-Cache technology.* *Note: AMD engineers performed extensive internal testing and validation for various applications using the X3D BIOS option found in AMD EPYC 9xx4X processors with AMD 3D V-Cache technology. This testing and validation cannot cover all applications or use cases. Testing has shown AMD 3D V-Cache to be beneficial for most workloads, however AMD recommends that you test and evaluate the benefits of enabling or disabling the X3D BIOS option for your application workloads in your environment and proceeding based on those results.*

Table 3-1: Processor core BIOS settings (Continued)

3.2 Power Efficiency Settings

Name	Default	Description
Power Profile Selection	Auto	• **Auto/0:** High-performance mode • **1:** Efficiency mode • **2:** Maximum I/O performance mode
Determinism Control	Auto	• **Auto:** Use default performance determinism settings. • **Manual:** Specify custom performance determinism settings.
Determinism Enable	Auto	• **Auto:** Performance. • **1:** Power.
TDP Control	Auto	• **Auto:** Use platform- and OPN-default TDP. • **Manual:** Set custom configurable TDP.
TDP	OPN Max	This option appears once the user sets the **TDP Control** to **Manual**. • **Values 85-400:** Set configurable TDP, in watts.
PPT Control	Auto	Enables or disables the **PPT** control. • **Auto:** Automatically set PPL in watts. • **Manual:** Specify a custom PPL.
PPT	OPN Max	This option appears once the user sets the **PPT Control** to **Manual**. • **Values 85-400:** Set configurable PPT, in watts.
CPPC	Auto	• **Enabled/Auto:** Allows the OS to make performance/power optimization requests using ACPI CPPC. • **Disabled:** Prevents the OS from making performance/power optimization requests using ACPI CPPC.

Table 3-2: Power efficiency BIOS settings

3.3 NUMA and Memory Settings

Name	Default	Description
LLC as NUMA Domain (ACPI SRAT L3 Cache as NUMA Domain)	Disabled	• **Disabled (recommended):** Both NUMA nodes (`cpubind`) and memory interleaving (`membind`) are determined by the NPS setting. • **Enabled:** Overrides the NPS setting for # of NUMA nodes by mapping each LLC as a NUMA node. This does not impact the memory interleaving
Nodes Per Socket (NPS)	1	**Memory Interleaving:** The **NPS** setting always determines the memory interleaving regardless of whether **LLC as NUMA** is **Enabled** or **Disabled**. # of NUMA nodes (if **LLC as NUMA Domain** is **Disabled**): • **NPS1/Auto:** One NUMA node per socket (Most cloud providers use this as it provides consistent average memory latency to all the accesses within a socket). • **NPS2:** Two NUMA nodes per socket. • **NPS4:** Four NUMA nodes per socket • **NPS0 (not recommended):** Only applicable for dual-socket systems. A single NUMA node is created for the whole two-socket platform. AMD recommends either NPS1 or NPS4 depending on your use case. **Windows systems:** Make sure that the number of logical processors per NUMA node is <=64. You can do this by using NPS2 or NPS4 instead of the default NPS1.
Memory Target Speed	Auto	• **Auto:** Determine the maximum memory speed based on SPD information from populated DIMMs and platform memory speed support. Alternatively, you can select: • **Values 3200–5600 MT/s:** Run the DRAM memory target clock speed at the specified speed. The DRAM memory target is the DDR rate. Your OEM system default value may vary.
Memory Interleaving	Auto	• **Auto/Enable:** Enables memory interleaving. • **Disable:** Allows for disabling memory interleaving. The N**UMA Nodes per Socket** setting will be honored regardless of this setting. AMD strongly recommends not disabling this setting because most production deployments benefit from memory interleaving.

Table 3-3: NUMA and memory BIOS settings

3.4 Infinity Fabric Settings

Name	Default	Description
3-4 xGMI Link Max Speed	Auto	• 12 Gbps • 16 Gbps • 17 Gbps • 18 Gbps • 20 Gbps • 22 Gbps • 23 Gbps • 24 Gbps • 25 Gbps/Auto • 26 Gbps • 27 Gbps • 28 Gbps • 30 Gbps • 32 Gbps Your OEM system default value may vary.
xGMI Link Width Control	Auto	• **Auto:** Use the default xGMI link width controller settings. • **Manual:** Specify a custom xGMI link width controller setting.
xGMI Force Link Width Control	Auto	• **Unforce:** Do not force the xGMI to a fixed width. • **Force:** Use the xGMI link to the user-specified width.
xGMI Force Link Width	Auto	• **0:** Force xGMI link width to x4. • **1:** Force xGMI link width to x8. • **2:** Force xGMI link width to x16.
xGMI Max Link Width Control	Auto	• **Auto:** Use the default xGMI link width controller settings. • **Manual:** Specify a custom xGMI link with controller setting.
xGMI Max Link Width	Auto	• **0:** Set max xGMI link width to x8. • **1:** Set max xGMI link width to x16.
APBDIS	Auto	• **0/Auto:** Dynamically switch the Infinity Fabric P-state based on link usage. • **1:** Enabled fixed Infinity Fabric P-state control.
DfPstate Range Support	Auto	• **Auto:** If this feature is enabled, the range value setting should follow the rule that MaxDfPstate<=MinDfPstate. Otherwise, it will not work. • **Enable:** Add the values MaxDfPstate & MinDfPstate. • **Disable:** No MaxDfPstate & MinDfPstate option.

Table 3-4: Infinity Fabric BIOS settings

DF C-States	Auto	Controls DF C-states. • **Disabled:** Prevents the AMD Infinity Fabric from entering a low-power state. • **Enabled/Auto:** Allows the AMD Infinity Fabric to enter a low-power state.

Table 3-4: Infinity Fabric BIOS settings (Continued)

3.5 PCIe, I/O, Security, and Virtualization Settings

Name	Default	Description
Local APIC Mode	Auto(0x02)	• **xAPIC:** Use xAPIC, supports up to 255 cores. • **x2APIC:** Supports more than 255 cores. • **Auto:** The system will choose the mode that best fits the number of active cores in the system. • **Compatibility:** Threads below 255 run in xAPIC with xAPIC ACPI structures, and threads 255 and above run in x2 mode with x2 ACPI structures. • **XApicMode (0x01):** Forces legacy xAPIC mode. • **X2ApicMode (0x02):** Forces x2APIC mode independent of thread count.
PCIe Speed PMM Control	Auto	• **0:** Dynamic link speed determined by power management functionality. • **1:** Static Target Link Speed (Gen4); sets the maximum idle link speed to 16 GT/s. • **Auto/2:** Static Target Link Speed (Gen5); sets the maximum idle link speed to 32 GT/s, thereby disabling the feature).
PCIe ARI Support (SRIOV)	Auto	• **Enabled/Auto:** Enables Alternative Routing ID interpretation. • **Disabled:** Disables Alternative Routing ID interpretation.
PCIe Ten Bit Tag Support	Auto	• **Enabled/Auto:** Enables PCIe 10-bit tags for supported devices. • **Disabled:** Disables PCIe 10-bit tags for all devices.
IOMMU	Auto	• **Enabled/Auto:** Enables IOMMU. AMD recommends setting this to `pt:pass-through` in the Linux kernel settings. • **Disabled:** Disables IOMMU.
AVIC	Disabled	Advanced Virtual Interrupt Controller. • **Disabled:** Disables AVIC. • **Enabled:** Enables AVIC.
x2AVIC	Disabled	x2AVIC is an extension of the advanced virtual interrupt controller. This feature currently requires a custom AMD Linux kernel. • **Disabled:** Disables x2AVIC. • **Enabled:** Enables x2AVIC.
TSME	Auto	• **Auto/Disabled:** Disables transparent secure memory encryption. • **Enabled:** Enables transparent secure memory encryption.
SEV	Disabled	In a multi-tenant environment (such as a cloud), Secure Encrypted Virtualization (SEV) mode isolates virtual machines from each other and from the hypervisor. • **Disabled:** SEV is disabled. • **Enabled:** SEV is enabled.

Table 3-5: PCIe, I/O, security, and virtualization BIOS settings

SEV-ES	Disabled	Secure Encrypted Virtualization-Encrypted State (SEV-ES) mode extends SEV protection to the contents of the CPU registers by encrypting them when a virtual machine stops running. Combining SEV and SEV-ES can reduce the attack surface of a VM by helping protect the confidentiality of data in memory. • **Disabled:** SEV-ES is disabled. • **Enabled:** SEV-ES is enabled.
SEV-SNP	Disabled	Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP) mode builds on SEV and SEV-ES by adding strong memory integrity protection to create an isolated execution environment that helps prevent malicious hypervisor-based attacks such as data replay and memory re-mapping. SEV-SNP also introduces several additional optional security enhancements that support additional VM use models, offer stronger protection around interrupt behavior, and increase protection against recently-disclosed side channel attacks. • **Disabled:** SEV-SNP is disabled. • **Enabled:** SEV-SNP is enabled.

Table 3-5: PCIe, I/O, security, and virtualization BIOS settings (Continued)

3.6 Higher-Level Settings

The system powers on to an initial state, after which succeeding software layers may affect system settings:

1. System firmware validates basic hardware functionality and allows users to change various settings via the BIOS Setup menus.
2. UEFI provides a shell environment that allows users to further interact with the system.
3. The operating system or hypervisor is the next software layer that provides control over system hardware.
4. Lastly, certain applications can also affect underlying hardware.

Each of the lines above may alter settings made by prior line, and some user changes require a reboot to take effect.

Please refer to your OEM documentation and/or applicable AMD Tuning Guide(s) for further guidance.

Chapter 4 BIOS Option Details

4.1 Processor Core Settings

4.1.1 Symmetric Multithreading (SMT) Settings

Enabling SMT causes neutral to negative performance impacts on some workloads, especially HPC. Also, some application licenses count the number of hardware threads enabled instead of the physical core count. It may therefore be best to disable SMT on your AMD EPYC 9004 Series Processor.

Some operating systems lack the x2APIC support required to support more than 384 threads. Disable SMT if you are running a non-x2APIC OS in a system with dual 96-core processors.

Setting	Options
SMT Control	• **Enable/Auto:** Two hardware threads per core. • **Disable:** Single hardware thread per core.

Table 4-1: SMT settings

4.1.2 Cache Prefetchers

Most workloads and production deployments benefit from the L1 & L2 Stream Hardware prefetchers gathering data and keeping the core pipeline busy, but some workloads that stress the memory bandwidth to its maximum capacity may perform better when some or all prefetchers are disabled. All prefetchers are enabled by default. Be sure to evaluate the prefetchers for your deployments.

Setting	Options
L1 Stream HW Prefetcher	This prefetcher uses the history of L1 cache memory access patterns to fetch additional sequential lines in ascending or descending order. • **Disable:** Disable prefetcher. • **Enable:** Enable prefetcher.
L1 Stride Prefetcher	The prefetcher uses the L1 cache memory access history of individual instructions to fetch additional lines when each access is a constant distance from the previous. • **Disable:** Disable prefetcher. • **Enable:** Enable prefetcher.
L1 Region Prefetcher	This prefetcher uses the L1 cache memory access history to fetch additional lines when the data access for a given instruction that tends to be followed by a consistent pattern of other accesses within a localized region. • **Disable:** Disable prefetcher. • **Enable:** Enable prefetcher.
L1 Burst Prefetch Mode	This prefetcher uses the L1 cache memory access history to fetch additional lines when the data access for a given instruction that tends to be followed by a consistent pattern of other accesses within a localized region. • **Disable:** Disable prefetcher. • **Enable:** Enable prefetcher.
L2 Stream HW Prefetcher	This prefetcher uses the history of L2 cache memory access patterns to fetch additional sequential lines in ascending or descending order. • **Disable:** Disable prefetcher. • **Enable:** Enable prefetcher.
L2 Up/Down Prefetcher	Uses the L2 cache memory access history to determine whether to fetch the next or previous line for all memory accesses. • **Disable:** Disable prefetcher. • **Enable:** Enable prefetcher.

Table 4-2: Cache prefetecher settings

4.1.3 Core Performance Boost

Core Performance Boost can be enabled or disabled. Enabling this setting allows the processor to opportunistically increase a set of CPU cores to higher than the CPU's rated base clock speeds based on the number of active cores, power, and thermal headroom in a system.

Some workloads don't need maximum core frequency to achieve acceptable performance. Limiting the maximum core boost frequency can reduce power consumption. The **BoostFmax** setting limits the maximum boost frequency but does not set a fixed frequency. The SoC will not exceed the maximum algorithm-allowable frequency if **BoostFmax** is set too high. Actual boost performance depends on many factors, including the other settings discussed in this tuning guide.

Setting	Options
Core Performance Boost	• **Enable/Auto:** Enables Core Performance Boost. • **Disable:** Disables Core Performance Boost.
BoostFmaxEn	• **Manual:** Use specified BoostFmax setting. • **Auto:** Use default BoostFmax setting.
BoostFmax	Specify the boost Fmax frequency limit to apply to all cores (MHz in decimal).

Table 4-3: Core boost settings

4.1.4 Global C-States Control

The **Global C-States Control** enables and disables C-states on the server across all cores. Disabling this feature means that the CPU cores can only be in C0 (active) or C1 state because the C1 state cannot be disabled. A CPU core will be in C1 state if the core is halted by the OS. IO based C-state generation and DF C-states include core processor C-States. If you have a low latency or extremely low jitter use case, then consider disabling DF C-states as described in this Tuning Guide. AMD strongly recommends not disabling Global C-states except for debugging.

Setting	Options
Global C-State Control	• **Enabled/Auto:** Controls IO based C-state generation and DF C-states, including core processor C-States • **Disabled:** AMD strongly recommends not disabling this option because this also disables core processor C-States.

4.1.5 AMD 3D V-Cache Technology

AMD EPYC 9xx4X Series Processors include AMD 3D V-Cache™ die stacking technology that enables denser, more efficient chiplet integration. AMD 3D Chiplet architecture stacks L3 cache tiles vertically to provide up to 96MB of L3 cache per die (and up to 1 GB L3 Cache per socket) while still providing socket compatibility with all AMD EPYC™ 9004 Series Processor models.

Setting	Options
X3D	• **Auto (Enabled):** Enables AMD 3D V-Cache technology on an AMD EPYC 9004X Series Processor, thereby increasing the total size of the L3 cache memory to 96MB per CCD. • **Disabled:** Disabling this option reduces the L3 cache in each CCD to 32MB.

4.2 Power Management Settings

4.2.1 Power Profile Selection

Setting	Options
Power Profile Selection Control	• **0:** High Performance mode (Default) • **1:** Efficiency Mode • **2:** Maximum IO Performance Mode

Table 4-4: Power profile selection

4.2.2 Power vs. Performance Determinism Settings

The **Determinism Enable** selects between:

- **Performance (default for most OPNs):** Uniform performance across identically configured systems in a datacenter. Set TDP and PPL to the same value, as described in "Processor Cooling and Power Dissipation Limit Settings" on page 26.
- **Power:** Maximum performance of any individual system by leveraging the capabilities of a given CPU to the maximum, resulting in a varying performance range across the datacenter or larger deployments.

Setting	Options
Determinism Control	• **Auto:** Use default performance determinism settings. • **Manual:** Specify custom power/performance determinism.
Determinism Enable	• **Auto:** This setting may be either **Power** or **Performance** based on OEM Platform and OPN selection. • **0:** Power (default) • **1:** Performance See "Processor Cooling and Power Dissipation Limit Settings" on page 26 for additional information.

Table 4-5: Power/performance settings

4.2.3 Processor Cooling and Power Dissipation Limit Settings

Thermal Design Power (TDP) allows modifying the CPU cooling limit and the Package Power Limit (PPL) allows modifying the CPU Power Dissipation Limit. Many platforms configure TDP to the maximum CPU-supported value. Most platforms also set the PPL to the same value as the TDP.

If you are using **Performance** determinism, then both the TDP and PPT must be set to the same value, as described in "Power vs. Performance Determinism Settings" on page 26. You can set the PPL to a lower value than the TDP to reduce system operating power. If you do this, then the CPU will control the Core Boost to keep the socket power dissipation at or below the PPL value.

If you are using **Power** determinism, then you can obtain maximum performance by setting the TDP and PPL to the maximum TDP value supported by the CPU. Setting the TDP and PPL to **Auto** sets both parameters to the CPU default TDP value for energy-efficient operation.

Setting	Options
TDP Control	• **Auto:** Use platform- and OPN-default TDP. • **Manual:** Set custom configurable TDP.
TDP	This option is available if the user sets the **TDP Control** to **Manual**. • **Values 85–400:** Set configurable TDP, in watts.
PPT Control	Enables or disables the **PPT** control. • **Auto:** Use platform- and OPN-default-PPL. • **Manual:** Set customized PPL.
PPT	This option is available if the user sets the **PPT Control** to **Manual**. • **Values 85–400:** Set PPT, in watts.

Table 4-6: TDP settings

4.2.4 ACPI—Collaborative Processor Performance Control (CPCC)

Enabling CPCC allows the OS to help maintain energy efficiency by controlling when and how much turbo can be applied. ACPI 5.0 introduced this feature. Not all operating systems support CPCC. Microsoft began supporting CPCC with Windows® Server® 2016.

Setting	Options
CPCC	• **Disabled:** Disabled. • **Enabled:** Allow the OS to make performance/power optimization requests using ACPI CPPC.

Table 4-7: CPCC settings

4.3 NUMA and Memory Settings

This section describes NUMA- and memory-related BIOS settings.

4.3.1 L3 Cache as NUMA Domain

This setting controls automatic or manual generation of distance information in the ACPI System Locality Information Table (SLIT) and NUMA proximity domains in the System Resource Affinity Table (SRAT). Some hypervisors and operating systems do not perform L3-aware scheduling, and some workloads will benefit from having the L3 declared as a NUMA domain. In dual-socket systems, the remote socket distance can affect memory allocation decisions. Setting this to a value of at least 32 (32 recommended) may improve scheduling of lightly-threaded workloads. Setting this to a value less than 32 (22 recommended) may improve scheduling of heavily-threaded workloads. In general:

- If a workload spans two sockets, then set the distance to < 32.
- If the workload can be confined to a socket, then set the distance to 32.

4.3.2 NUMA Nodes per Socket (NPS)

This setting enables a trade-off between minimizing local memory latency for NUMA-aware or highly parallelizable workloads vs. maximizing per-core memory bandwidth for non-NUMA-friendly workloads. NPS2 and/or NPS4 may not be an option on certain OPNs or with certain memory populations.

- **NPS1:** Indicates a single NUMA node per socket. This setting configures all memory channels on the processor into a single NUMA domain. All of the processor cores, all attached memory, and all PCIe devices connected to the SoC are in that one NUMA domain. Memory accesses are interleaved across all 24 memory channels into a single address space. The default configuration (one NUMA domain per socket) is recommended for most workloads.

- **NPS2:** 2 NUMA domains per socket, which interleaves the corresponding six memory channels within the same 6 CCD NUMA domain. Half of the cores and half of the memory channels of the SoC are grouped together into one NUMA domain, with the remaining cores and memory channels grouped into the second NUMA domain. Memory is interleaved across the six memory channels of each NUMA domain.

- **NPS4:** 4 partitions the processor into four NUMA domains with each logical quadrant configured as its own NUMA domain. Memory is interleaved across the three memory channels within each quadrant. PCIe devices will be local to one of the four NUMA domains depending on the quadrant (of the I/O die) that has the PCIe root complex for that device. Every pair of memory channels is interleaved. This is recommended for HPC and other highly-parallel workloads. You must use NPS4 when booting Windows systems with SMT enabled for AMD EPYC processors with more than 64 cores because Windows limits the size of a CPU group to a maximum of 64 logical cores.

Note: For Windows systems, verify that the number of logical processors per NUMA node <=64 by using either NPS2 or NPS4 instead of the default NPS1.

Enabling **ACPI SRAT L3 Cache as NUMA Domain** (another name for **L3 as NUMA**) determines the number of NUMA nodes and overrides the number of NUMA nodes specified by the NPS setting while still using the NPS setting to determine the memory interleaving granularity.

Setting	Options
L3 Cache as NUMA Domain (ACPI SRAT L3 Cache as NUMA Domain)	• **Disabled/Auto:** Do not report each L3 cache to the OS as a NUMA domain. • **Enabled:** Report each L3 cache to the OS as a NUMA domain.
NUMA Node Per Socket	• **NPS0:** Interleave memory accesses across all channels in both sockets (not recommended). • **NPS1:** Interleaves memory accesses across all channels in each socket and report one NUMA node per socket unless **L3 Cache as NUMA** is enabled. • **NPS2:** Interleaves memory accesses the channels associated with each half of a socket and reports two NUMA nodes per socket unless **L3 Cache as NUMA** is enabled. • **NPS4:** Interleaves memory accesses across the channels associated with a quadrant of each socket and reports four NUMA nodes per socket unless **L3 Cache as NUMA** is enabled.

Table 4-8: NPS settings

4.3.3 Memory Target Speed

By default, the 4th Gen AMD EPYC processor BIOS runs at the maximum target frequency allowed by the platform and DIMM. This configuration allows maximum memory bandwidth and lowest latency for the processor. Lowering the memory clock speed reduces memory controller power consumption and allows the rest of the SoC to consume more power, thereby potentially boosting performance elsewhere for certain workloads.

Setting	Options
Memory Target Speed	• **Auto:** Determine maximum memory speed based on SPD information from populated DIMMs and platform memory speed support. • **Values 3200–5600 MT/s:** Run the DRAM memory target speed at the specified speed (the DRAM memory target is the DDR rate.)

Table 4-9: Memory clock settings

4.3.4 Memory Interleaving

This setting allows you to enable or disable memory interleaving within a NUMA node. The NUMA Nodes per Socket (NPS) setting will be honored regardless of this setting. This BIOS setting does not impact the number of NUMA nodes or how memory channels are mapped to the NUMA nodes.

Note: AMD strongly recommends not disabling this setting because most applications and deployments benefit from memory interleaving.

Setting	Options
Memory Interleaving	• **Auto/Enable:** Enables memory interleaving. • **Disable:** Allows for disabling memory interleaving. The NUMA Nodes per Socket (NPS) setting will be honored regardless of this setting.

Table 4-10: Memory interleaving settings

4.4 Infinity Fabric Settings

This section discusses BIOS settings related to AMD Infinity Fabric technology.

4.4.1 Link Speed

Lowering the link speed decreases cross-socket bandwidth and increases cross-socket latency but can also save uncore power (CPU power not consumed by the cores) to either:

- Increase core frequency.
- Reduce overall power consumption.

Setting	Options
3-4 Link xGMI Max Speed	• **25 Gbps/Auto** • Additional options depend on the OEM platform: 17, 18, 20, 22, 23, 24, 25, 26, 27, 28, 30, 32 Gbps

Table 4-11: Link speed settings

4.4.2 xGMI Link Width Management

xGMI Dynamic Link Width Management saves power during periods of low socket-to-socket data traffic by reducing the number of active xGMI lanes per link from 16 to 8, or x4 if the xGMI links have limited traffic. Latency may increase in some scenarios involving low-bandwidth, latency-sensitive traffic as the processor transitions from a low-power xGMI state to full-power xGMI state. Setting **xGMI Link Width Control** to **Manual** and specifying a **Force Link Width** eliminates any such latency jitter. Applications that are not sensitive to both socket-to-socket bandwidth and latency can use a forced link width of 8 (or 2 on certain platforms) to save power, which can divert more power to the cores for boost.

Setting	Options
xGMI Link Max Speed	NUMA-unaware workloads may need maximum xGMI bandwidth because of extensive cross-socket communications. NUMA-aware workloads may want to minimize xGMI power because they do not have a lot of cross socket traffic and prefer to use the increased CPU boost. • 25 Gbps/Auto • 32 Gbps
xGMI Link Width Control	• **Auto:** Hide the **Max Link Width** and **Force Link Width** control options. • **Manual:** Show **Max Link Width** and **Force Link Width** control options.
xGMI Max Link Width	• **0:** Max width x8, min width x8 (x4 on certain platforms). • **1:** Max width x16, min width x8 (x4 on certain platforms).
xGMI Max Link Width Control	• **Auto:** Hide the **xGMI Max Link Width** control. • **Manual:** Show the **xGMI Max Link Width** control.
xGMI Force Link Width Control	• **Unforce:** Use automatic xGMI Link Width selection. • **Force:** Use the xGMI Force Link Width link width.
xGMI Force Link Width	• **0:** Use width x4 (not supported on all platforms). • **1:** Use width x8.

Table 4-12: DLWM settings

4.4.3 Power States

Enable or disable Algorithm Performance Boost (APB). By default, the AMD Infinity Fabric selects between a full- and low-power fabric clock and memory clock based on usage. Latency may increase in some scenarios involving low-bandwidth, latency-sensitive traffic as the processor transitions from low to full power. Setting **APBDIS** to 1 (APB disabled) and specifying a fixed Infinity Fabric P-state of 0 forces the AMD Infinity Fabric and memory controllers into full-power mode and significantly reduces latency jitters.

Setting	Options
APB Disable (APBDIS)	• **0:** Dynamically switch Infinity Fabric P-state based on link usage. • **1:** Enable fixed Infinity Fabric P-state control.
DfPstate	DfPstate index to set below when APBDIS [1]. • **Min Value:** 0 (default); highest-performing AMD Infinity Fabric P-state. • **Max Value:** 2 • **Pn:** Next-highest-performing AMD Infinity Fabric P-state.
DfPstate Range Support	DF Pstate selection is overridden by the APB_DIS BIOS option if it is selected. If this feature is enabled, then range value setting should follow the rule that MaxDfPstate<=MinDfPstate. Otherwise, it will not work.

Table 4-13: Power state settings

4.4.4 DF C-States

Much like CPU cores, the AMD Infinity Fabric can enter lower-power states while idle, but a delay occurs when transitioning back to full-power mode that causes some latency jitter. Disabling this feature for workloads requiring low latency and/or bursty I/O will increase both performance and power consumption.

Setting	Options
DF C-states	• **Auto/Enabled:** Allow the AMD Infinity Fabric to enter a low-power state. • **Disabled:** Prevent the AMD Infinity Fabric from entering a low-power state.

Table 4-14: C-state settings

4.5 PCIe, I/O, Security, and Virtualization Settings

4.5.1 APIC Settings

Interrupt delivery is generally faster when using x2APIC compared to the legacy xAPIC mode, but not all operating systems include AMD x2APIC support. AMD recommends this mode if your OS supports it, including for configurations with fewer than 256 logical processors.

Setting	Options
Local APIC Mode	• **APIC:** Use xAPIC, which supports up to 255 cores. • **x2APIC:** Supports more than 255 cores. • **Auto:** The system will choose the mode that best fits the number of active cores in the system. • **Compatibility:** Threads below 255 run in xAPIC with xAPIC ACPI structures, and threads 255 and above run in x2 mode with x2 ACPI structures. • **XApicMode (0x01):** Force legacy xApic mode • **X2ApicMode (0x02):** F force x2Apic mode independent of thread count.

Table 4-15: APIC settings

4.5.2 PCIe Speed PMM Control

The **PCIe Speed PMM Control** is an activity-based power management feature designed for PCIe Gen5 endpoints. After a device is trained, the controller monitors activity and adjusts the link speed accordingly. An idle PCIe Gen5 link will be turned down to the next highest available device speed and will return to the Gen5 speed when activity increases.

Note: PCIe Gen5 devices that train using Equalization Bypass do not enable Gen3 or Gen4 speed. When idle, they will operate at Gen2 speed. Gen5 devices that train with full equalization support all speeds and will be turned down to Gen4 speed when idle.

Setting	Options
PCIe PMM Speed Control	• **0:** Dynamic link speed determined by power management functionality. • **1:** Static Target Link Speed (Gen4); sets the maximum idle link speed to 16 GT/s. • **Auto/2:** Static Target Link Speed (Gen5); sets the maximum idle link speed to 32 GT/s disabling the feature.

Table 4-16: PCIe Speed PMM Control settings

4.5.3 SR-IOV Settings

SR-IOV requires enabling PCIe Alternative Routing-ID interpretation (ARI) on both root complexes and endpoints. ARI devices interpret the PCI address as an 8-bit function number instead of a 3-bit function number, and the device number is implied to be 0.

Setting	Options
PCIe ARI Support [SRIOV]	• **Disable:** Disable Alternative Routing ID interpretation. • **Enable:** Enable Alternative Routing ID interpretation.

Table 4-17: SR-IOV settings

4.5.4 PCIe Ten Bit Tag

A PCIe adapter must support 10-bit extended tags to achieve maximum PCIe Gen 5 bandwidth. This boosts adapter performance by allowing a 3x increase over the previous number of non-posted requests. Not all PCIe Gen 5 devices support 10-bit extended tags, which can cause issues during boot. Disabling this feature allows the server to boot if the adapter is having issues.

Setting	Options
PCIe Ten Bit Tag Support	• **Auto/Enable:** Enable PCIe 10-bit tags for supported devices. • **Disable:** Disable PCIe 10-bit tags for all devices.

Table 4-18: PCIe 10-bit settings

4.5.5 Input-Output Memory Management Unit (IOMMU) Settings

Enabling the IOMMU allows devices such as the AMD EPYC processor-integrated SATA controller to present separate IRQs for each attached device instead of one IRQ for the subsystem. The IOMMU also allows operating systems to provide additional protection for DMA capable I/O devices. If you believe the IOMMU is limiting performance, then leave it enabled in BIOS and disable it via OS options (e.g., `iommu=pt` on the Linux® kernel command line). Enabling IOMMU is required when using x2APIC.

Setting	Options
IOMMU	• **Enabled:** Enable IOMMU. • **Disabled:** Disable IOMMU.

Table 4-19: IOMMU settings

4.5.6 Transparent Secure Memory Encryption (TSME)

This feature provides hardware memory encryption of all data stored on system DIMMs that is invisible to the OS and slightly increases memory latency.

Setting	Options
TSME	• **Auto / Disabled:** Disable transparent secure memory encryption. • **Enabled:** Enable transparent secure memory encryption.

Table 4-20: TSME settings

4.5.7 SEV, SEV-ES, and SEV-SNP

Please see the *AMD EPYC™ 9004 Cloud Infrastructure and Datacenter Design & Configuration Guide* (available from AMD EPYC Tuning Guides) for information about AMD EPYC security features.

4.5.8 AVIC& x2AVIC

Enabling Advanced Virtual Interrupt Controller (AVIC) may not improve system performance compared to systems with AVIC disabled. In some cases, performance with AVIC enabled may be significantly lower than system performance with AVIC disabled. AMD recommends not enabling AVIC on production systems.

x2AVIC is an extension of the advanced virtual interrupt controller (AVIC) that supports more than 255 virtual CPUs and offers better performance than AVIC. It can also be thought of as hardware virtualization of x2APIC. With x2AVIC, the guest's local APIC hardware-assisted virtualization extends to 512 virtual CPUs. Before AMD Socket SP5 Processors, the VM needed to disable x2APIC capabilities because using them would bypass hardware AVIC and use software-emulated x2APIC. However, with x2AVIC, the guest will be able to leverage x2APIC performance advantages.

Name	Default	Description
AVIC	Disabled	Advanced Virtual Interrupt Controller. • **Disabled:** Disables AVIC. • **Enabled:** Enables AVIC.
x2AVIC	Disabled	x2AVIC is an extension of the advanced virtual interrupt controller. This feature currently requires a custom AMD Linux kernel. • **Disabled:** Disables x2AVIC. • **Enabled:** Enables x2AVIC.

Table 4-21: AVIC and x2AVIC settings

4.6 Options Eliminated in AMD EPYC 9004

The following BIOS options were available in 3rd Gen AMD EPYC processors but are not available for user customization with AMD EPYC 9004 Series Processors as part of architecture improvements:

- EfficiencyModeEn
- Root Complex LCLK Frequency
- Preferred I/O
- Preferred I/O Bus

Chapter 5 Workload-Specific BIOS Settings

Use these guidelines for general-purpose workloads. Some cases list the benchmarks used in order to better describe the workloads used to obtain the recommended settings. Default settings are used when labeled default.

5.1 General-Purpose Workloads

5.1.1 Processor Core Settings

Setting	CPU Intensive	Java Throughput	Java Latency	Power Efficiency
SMT Control	*default*	*default*	*default*	*default*
L1 Stream HW Prefetcher	*default*	*default*	*default*	*default*
L1 Stride Prefetcher	*default*	*default*	*default*	*default*
L1 Region Prefetcher	*default*	*default*	*default*	*default*
L1 Burst Prefetch Mode	*default*	*default*	*default*	*default*
L2 Stream HW Prefetcher	*default*	*default*	*default*	*default*
L2 Up/Down Prefetcher	*default*	*default*	*default*	*default*
BoostFmaxEn	*default*	*default*	*default*	*default*
BoostFmax	*default*	*default*	*default*	*default*

Table 5-1: Processor core settings

5.1.2 Power Management Settings

Setting	CPU Intensive	Java Throughput	Java Latency	Power Efficiency
Power Profile Selection	High Performance	High Performance	High Performance	Efficiency Mode
Determinism Control	*default*	*default*	*default*	*default*
Determinism Enable	*default*	*default*	*default*	*default*
TDP Control	Manual	Manual	Manual	Manual
TDP	OPN Max	OPN Max	OPN Max	OPN Max
PPT Control	Manual	Manual	Manual	Manual
PPT	OPN Max	OPN Max	OPN Max	OPN Max
CPPC	*default*	*default*	*default*	*default*

Table 5-2: Power efficiency settings

5.1.3 NUMA and Memory Settings

Setting	CPU Intensive	Java Throughput	Java Latency	Power Efficiency
ACPI SRAT L3 Cache as NUMA Domain	*default*	*default*	*default*	Enabled
NUMA Nodes per Socket (NPS)	*default*	4	2	4
Memory Target Speed	4800	4800	4800	4800
Memory Interleaving	*default*	*default*	*default*	*default*

Table 5-3: NUMA and memory settings

5.1.4 Infinity Fabric Settings

Setting	CPU Intensive	Java Throughput	Java Latency	Power Efficiency
xGMI Link Max Speed	*default*	*default*	*default*	*default*
xGMI Link Width Control	*default*	*default*	*default*	*default*
xGMI Max Link Width	*default*	*default*	*default*	*default*
xGMI Max Link Width Control	*default*	*default*	*default*	*default*
xGMI Force Link Width Control	*default*	*default*	*default*	*default*
xGMI Force Link Width	*default*	*default*	*default*	*default*
APBDIS	*default*	*default*	*default*	1
DF C-States	*default*	*default*	*default*	*default*

Table 5-4: Infinity Fabric DP settings

Note: NUMA-unaware workloads may need maximum xGMI bandwidth because of extensive cross-socket communications. NUMA-aware workloads may want to minimize xGMI power because they do not have a lot of cross socket traffic and prefer to use the increased CPU boost.

5.1.5 I/O Settings

Setting	CPU Intensive	Java Throughput	Java Latency	Power Efficiency
Local APIC Mode	*default*	*default*	*default*	*default*
PCIe Speed PMM Control	*default*	*default*	*default*	*default*
PCIe ARI Support [SRIOV]	*default*	*default*	*default*	*default*
PCIe Ten Bit Tag Support	*default*	*default*	*default*	*default*
IOMMU	*default*	*default*	*default*	*default*
TSME	*default*	*default*	*default*	*default*

Table 5-5: I/O settings

5.2 Memory and I/O Intensive Workloads

5.2.1 Processor Core Settings

Setting	Memory Throughput	Storage I/O Throughput	NIC Throughput	NIC Latency
SMT Control	*default*	*default*	Disabled	Disabled
L1 Stream HW Prefetcher	*default*	*default*	*default*	*default*
L1 Stride HW Prefetcher	*default*	*default*	*default*	*default*
L1 Region Prefetcher	*default*	*default*	*default*	*default*
L2 Stream HW Prefetcher	*default*	*default*	*default*	*default*
L2 Stream HW Prefetcher	*default*	*default*	*default*	*default*
L2 Up/Down Prefetcher	*default*	*default*	*default*	*default*
BoostFmaxEn	*default*	*default*	*default*	*default*
BoostFmax	*default*	*default*	*default*	*default*

Table 5-6: Processor core settings

5.2.2 Power Management Settings

Setting	Memory Throughput	Storage I/O Throughput	NIC Throughput	NIC Latency
Power Profile Selection	*default*	*default*	*default*	*default*
Determinism Control	*default*	Enabled	Manual	Manual
Determinism Enable	*default*	Power	Performance	Performance
TDP Control	*default*	*default*	*default*	*default*
TDP	OPN Max	*default*	*default*	*default*
PPT Control	*default*	*default*	*default*	*default*
PPT	OPN Max	*default*	*default*	*default*
CPPC	*default*	*default*	*default*	*default*

Table 5-7: Power efficiency settings

5.2.3 NUMA and Memory Settings

Setting	Memory Throughput	Storage I/O Throughput	NIC Throughput	NIC Latency
ACPI SRAT L3 Cache as NUMA Domain	Enabled	*default*	*default*	*default*
NUMA Nodes per Socket (NPS)	*default*	*default*	*default*	*default*
Memory Target Speed	4800	*default*	*default*	*default*
Memory Interleaving	*default*	*default*	*default*	*default*

Table 5-8: NUMA and memory settings

5.2.4 Infinity Fabric Settings

Setting	Memory Throughput	Storage I/O Throughput	NIC Throughput	NIC Latency
xGMI Link Max Speed	*default*	*default*	*default*	*default*
xGMI Link Width Control	*default*	*default*	*default*	*default*
xGMI Max Link Width	*default*	*default*	*default*	*default*
xGMI Max Link Width Control	*default*	*default*	*default*	x16
xGMI Force Link Width Control	*default*	*default*	*default*	*default*
xGMI Force Link Width	*default*	*default*	*default*	*default*
APBDIS	*default*	1	Disabled	Disabled
DF C-States	*default*	*default*	*default*	*default*

Table 5-9: Infinity Fabric settings

Note: NUMA-unaware workloads may need maximum xGMI bandwidth because of extensive cross-socket communications. NUMA-aware workloads may want to minimize xGMI power because they do not have a lot of cross socket traffic and prefer to use the increased CPU boost.

5.2.5 I/O Settings

Setting	Memory Throughput	Storage I/O Throughput	NIC Throughput	NIC Latency
Local APIC Mode	*default*	*default*	*default*	*default*
PCIe Speed PMM Control	*default*	Static TLS Gen5	Static TLS Gen5	Static TLS Gen5
PCIe ARI Support [SRIOV]	*default*	*default*	*default*	*default*
PCIe Ten Bit Tag Support	*default*	*default*	*default*	*default*
IOMMU	*default*	*default*	*default*	*default*
TSME	*default*	*default*	*default*	*default*

Table 5-10: I/O settings

5.3 Virtualization and Containers

5.3.1 Processor Core Settings

Setting	VMware vSphere Optimized	Linux KVM Optimized	Containers
SMT Control	Enabled	Enabled	Enabled
L1 Stream HW Prefetcher	*default*	*default*	*default*
L1 Stride HW Prefetcher	*default*	*default*	*default*
L1 Region Prefetcher	*default*	*default*	*default*
L1 Burst Prefetch Mode	*default*	*default*	*default*
L2 Stream HW Prefetcher	*default*	*default*	*default*
L2 Up/Down Prefetcher	*default*	*default*	*default*
BoostFmaxEn	*default*	*default*	*default*
BoostFmax	*default*	*default*	*default*

Table 5-11: Processor core settings

5.3.2 Power Management Settings

Setting	VMware vSphere Optimized	Linux KVM Optimized	Containers
Power Profile Selection	*default*	*default*	*default*
Determinism Control	*default*	Enabled	*default*
Determinism Enable	*default*	Performance	*default*
TDP Control	*default*	*default*	*default*
TDP	*default*	*default*	*default*
PPT Control	*default*	*default*	*default*
PPT	*default*	*default*	*default*
CPPC	*default*	*default*	*default*

Table 5-12: Power efficiency settings

5.3.3 NUMA and Memory Settings

Setting	VMware vSphere Optimized	Linux KVM Optimized	Containers
ACPI SRAT L3 Cache as NUMA Domain	*default*	Enabled	*default*
NUMA Nodes per Socket (NPS)	*default*	*default*	*default*
Memory Target Speed	*default*	*default*	*default*
Memory Interleaving	*default*	*default*	*default*

Table 5-13: NUMA and memory settings

5.3.4 Infinity Fabric Settings

Setting	VMware vSphere Optimized	Linux KVM Optimized	Containers
xGMI Link Max Speed	*default*	*default*	*default*
xGMI Link Width Control	*default*	*default*	*default*
xGMI Max Link Width	*default*	*default*	*default*
xGMI Max Link Width Control	*default*	*default*	*default*
xGMI Force Link Width Control	*default*	*default*	*default*
xGMI Force Link Width	*default*	*default*	*default*
APBDIS	*default*	*default*	*default*
DF C-states	*default*	*default*	*default*

Table 5-14: Infinity Fabric settings

Note: NUMA-unaware workloads may need maximum xGMI bandwidth because of extensive cross-socket communications. NUMA-aware workloads may want to minimize xGMI power because they do not have a lot of cross socket traffic and prefer to use the increased CPU boost.

5.3.5 I/O Settings

Setting	VMware vSphere Optimized	Linux KVM Optimized	Containers
Local APIC Mode	*default*	*default*	*default*
PCIe Speed PMM Control	*default*	*default*	*default*
PCIe ARI Support [SRIOV]	*default*	*default*	*default*
PCIe Ten Bit Tag Support	*default*	*default*	*default*
IOMMU	*default*	*default*	*default*
TSME	*default*	*default*	*default*

Table 5-15: I/O settings

5.4 Database and Analytics

5.4.1 Processor Core Settings

Setting	RDBMS Optimized	Big Data Analytics Optimized	IoT Gateway
SMT Control	*default*	*default*	*default*
L1 Stream HW Prefetcher	*default*	*default*	*default*
L1 Stride HW Prefetcher	*default*	*default*	*default*
L1 Region Prefetcher	*default*	*default*	*default*
L1 Burst Prefetch Mode	*default*	*default*	*default*
L2 Stream HW Prefetcher	*default*	*default*	*default*
L2 Up/Down Prefetcher	*default*	*default*	*default*
BoostFmaxEn	*default*	*default*	*default*
BoostFmax	*default*	*default*	*default*

Table 5-16: Processor core settings

5.4.2 Power Management Settings

Setting	RDBMS Optimized	Big Data Analytics Optimized	IoT Gateway
Power Profile Selection	Maximum IO Performance	*default*	*default*
Determinism Control	Enabled	*default*	*default*
Determinism Enable	Power	*default*	*default*
TDP Control	Manual	*default*	*default*
TDP	OPN Max	*default*	*default*
PPT Control	Manual	*default*	*default*
PPT	OPN Max	*default*	*default*
CPPC	*default*	*default*	*default*

Table 5-17: Power efficiency settings

5.4.3 NUMA and Memory Settings

Setting	RDBMS Optimized	Big Data Analytics Optimized	IoT Gateway
ACPI SRAT L3 Cache as NUMA Domain	*default*	Enabled	*default*
NUMA Nodes per Socket (NPS)	• NPS4 (Windows) • *default* (Linux)	*default*	*default*
Memory Target Speed	Auto	Auto	Auto
Memory Interleaving	*default*	*default*	*default*

Table 5-18: NUMA and memory settings

5.4.4 Infinity Fabric Settings

Setting	RDBMS Optimized	Big Data Analytics Optimized	IoT Gateway
xGMI Link Max Speed	*default*	*default*	*default*
xGMI Link Width Control	*default*	*default*	*default*
xGMI Max Link Width	*default*	*default*	*default*
xGMI Max Link Width Control	*default*	*default*	*default*
xGMI Force Link Width Control	*default*	*default*	*default*
xGMI Force Link Width	*default*	*default*	*default*
APBDIS	1	*default*	*default*
DF C-States	*default*	*default*	*default*

Table 5-19: Infinity Fabric settings

Note: NUMA-unaware workloads may need maximum xGMI bandwidth because of extensive cross-socket communications. NUMA-aware workloads may want to minimize xGMI power because they do not have a lot of cross socket traffic and prefer to use the increased CPU boost.

5.5 I/O Settings HPC and Telco Settings

Setting	RDBMS Optimized	Big Data Analytics Optimized	IoT Gateway
Local APIC Mode	*default*	*default*	*default*
PCIe Speed PMM Control	*default*	*default*	*default*
PCIe ARI Support [SRIOV]	*default*	*default*	*default*
PCIe Ten Bit Tag Support	*default*	*default*	*default*
IOMMU	*default*	*default*	*default*
TSME	*default*	*default*	*default*

Table 5-20: I/O settings

5.5.1 Processor Core Settings

Setting	HPC	OpenStack® NFV	OpenStack® for RealTime Kernel (NFV)	EDA
SMT Control	Disabled	*default*	*default*	Disabled
L1 Stream HW Prefetcher	*default*	*default*	*default*	*default*
L1 Stride HW Prefetcher	*default*	*default*	*default*	*default*
L1 Region Prefetcher	*default*	*default*	*default*	*default*
L1 Burst Prefetch Mode	*default*	*default*	*default*	*default*
L2 Stream HW Prefetcher	*default*	*default*	*default*	*default*
L2 Up/Down Prefetcher	*default*	*default*	*default*	*default*
BoostFmaxEn	*default*	*default*	*default*	*default*
BoostFmax	*default*	*default*	*default*	*default*

Table 5-21: Processor core settings

5.5.2 Power Management Settings

Setting	HPC	OpenStack® NFV	OpenStack® for RealTime Kernel (NFV)	EDA
Power Profile Selection	High Performance	*default*	*default*	*default*
Determinism Control	*default*	Manual	Manual	Manual
Determinism Enable	Performance	Performance	Performance	Power
TDP Control	Manual	*default*	*default*	Manual
TDP	OPN Max	*default*	*default*	OPN Max
PPL Control	Manual	*default*	*default*	Manual
PPL	OPN Max	*default*	*default*	OPN Max
CPPC	*default*	*default*	*default*	*default*

Table 5-22: Power efficiency settings

5.5.3 NUMA and Memory Settings

Setting	HPC	OpenStack® NFV	OpenStack® for RealTime Kernel (NFV)	EDA
ACPI SRAT L3 Cache as NUMA Domain	*default*	*default*	*default*	*default*
NUMA Nodes per Socket (NPS)	4	*default*	*default*	2
Memory Target Speed	*default*	*default*	*default*	*default*
Memory Interleaving	*default*	*default*	*default*	*default*

Table 5-23: NUMA and memory settings

5.5.4 Infinity Fabric Settings

Setting	HPC	OpenStack® NFV	OpenStack® for RealTime Kernel (NFV)	EDA
xGMI Link Max Speed	*default*	*default*	*default*	*default*
xGMI Link Width Control	*default*	*default*	*default*	Manual
xGMI Max Link Width	*default*	*default*	*default*	*default*
xGMI Max Link Width Control	*default*	*default*	*default*	Manual
xGMI Force Link Width Control	*default*	*default*	*default*	Force
xGMI Force Link Width	*default*	*default*	*default*	*default*
APBDIS	*default*	*default*	*default*	1
DF C-States	*default*	*default*	*default*	Disabled

Table 5-24: Infinity Fabric settings

Note: NUMA-unaware workloads may need maximum xGMI bandwidth because of extensive cross-socket communications. NUMA-aware workloads may want to minimize xGMI power because they do not have a lot of cross socket traffic and prefer to use the increased CPU boost.

5.5.5 I/O Settings

Setting	HPC	OpenStack® NFV	OpenStack® for RealTime Kernel (NFV)	EDA
Local APIC Mode	*default*	*default*	*default*	*default*
PICe Speed PMM Control	*default*	*default*	*default*	*defaultdefault*
PCIe ARI Support [SRIOV]	*default*	*default*	*default*	*default*
PCIe 10-Bit Tag Support	*default*	*default*	*default*	*default*
IOMMU	Enable*	*default*	*default*	*default* (Enabled with 2P 64-core CPUs with SMT=Enabled)
TSME	*default*	*default*	*default*	*default*

Table 5-25: I/O settings

* = For HPC, enable IOMMU in BIOS. Within Linux, add the boot command `iommu=pt` to set the IOMMU to passthrough mode.

This page intentionally left blank.

Chapter 6

Processor Identification

Figure 6-1 shows the processor naming convention for AMD EPYC 9004 Series Processors and how to use this convention to identify particular processors models:

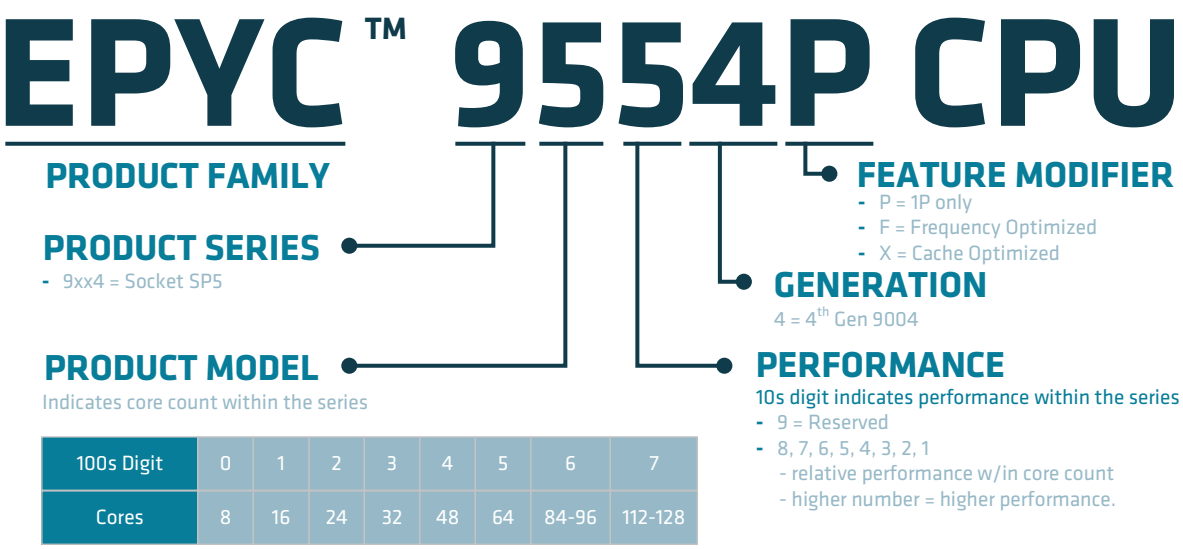

Figure 6-1: AMD EPYC SoC naming convention

6.1 CPUID Instruction

Software uses the `CPUID` instruction (`Fn0000_0001_EAX`) to identify the processor and will return the following values:

- **Family:** 19h identifies the "Zen 4" architecture
- **Model:** Varies with product. For example, EPYC Family 19h, Model 10h corresponds to an "A" part "Zen 4" CPU.
  - **91xx-96xx (including "X" OPNs):** Family 19h, Model 10-1F
  - **97xx:** Family 19h, Model A0-AF
- **Stepping:** May be used to further identify minor design changes

For example, `CPUID` values for Family, Model, and Stepping (decimal) of 25, 17, 1 correspond to a "B1" part "Zen 4" CPU.

6.2 New Software-Visible Features

AMD EPYC 9004 Series Processors introduce several new features that enhance performance, ISA updates, provide additional security features, and improve system reliability and availability. Some of the new features include:

- 5-level Paging
- AVX-512 instructions on a 256-byte datapath, including BFLOAT16 and VNNI support.
- Fast Short Rep STOSB and Rep CMPSB

Not all operating systems or hypervisors support all features. Please refer to your OS or hypervisor documentation for specific releases to identify support for these features.

Please also see the latest version of the *AMD64 Architecture Programmer's Manuals* or *Processor Programming Reference (PPR) for AMD Family 19h*.

6.2.1 AVX-512

AVX-512 is a set of individual instructions supporting 512-bit register-width data (i.e., single instruction, multiple data [SIMD]) operations. AMD EPYC 9004 Series Processors implement AVX 512 by "double-pumping" 256-bit-wide registers. AMD's AVX-512 design uses the same 256-bit data path that exists throughout the Zen4 core and enables the two parts to execute on sequential clock cycles. This means that running AVX-512 instructions on AMD EPYC 9004 Series will cause neither drops on effective frequencies nor increased power consumption. On the contrary, many workloads run more energy-efficiently on AVX-512 than on AVX-256P.

Other AVX-512 support includes:

- Vectorized Neural Network Instruction (VNNI) instructions that are used in deep learning models and accelerate neural network inferences by providing hardware support for convolution operations.
- Brain Floating Point 16-bit (BFLOAT16) numeric format. This format is used in Machine Learning applications that require high performance but must also conserve memory and bandwidth. BFLOAT16 support doubles the number of SIMD operands over 32-bit single precision FP, allowing twice the amount of data to be processed using the same memory bandwidth. BFLOAT16 values mantissa dynamic range at the expense of one radix point.

Chapter 7 Debugging BIOS Setting Changes

Some BIOS default settings can be changed from the command line by a user with adequate privileges. Some of these settings take effect immediately while others may require rebooting. Some settings may not be available in the BIOS but can be set at the OS level. Some OS-level settings can either override or modify the expected BIOS default behavior. This section lists some of the settings that could have a significant impact on system performance and response times.

Note: For each of the following examples, if you are running Windows, then please see the Microsoft® Windows® Server Tuning Guide for AMD EPYC™ 9004 Series Processors (available from AMD EPYC Tuning Guides).

NUMA Node Configurations:

The number of NUMA nodes, association of nodes to the memory channels, and binding of processes to specific NUMA nodes and memory nodes plays a vital role in many deployments. The **NPS** and **L3 as NUMA** settings control the number of NUMA nodes and memory channels associated with the NUMA nodes. To do this, a user leverages the NUMA command options as defined for a given OS and platforms and expects the processes to bind to certain CPUs and memory as per default BIOS settings. However, many OS-level commands and daemons may, if enabled, alter this expected behavior. The following methodology is recommended to understand and debug NUMA related issues:

Understand NUMA topology:

Verify that the number of NUMA nodes and topology are correct.

Linux has many commands such as `lstopo`, `hwloc`, and `numactl`. Here is some sample `numactl` output from a single socket AMD EPYC OPN with 12CCDs per Socket where NPS=4 and L3 as NUMA=Disabled:

```
$ numactl –H
available: 4 nodes (0-3)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 96 97 98 99 100
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
node 0 size: 32225 MB
node 0 free: 31374 MB
node 1 cpus: 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 120 121
122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
node 1 size: 64448 MB
node 1 free: 63633 MB
node 2 cpus: 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 144 145
146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167
node 2 size: 96407 MB
node 2 free: 95509 MB
node 3 cpus: 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 168 169
170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
node 3 size: 32164 MB
node 3 free: 31221 MB
node distances:
node   0   1   2   3
   0:  10  12  12  12
   1:  12  10  12  12
   2:  12  12  10  12
   3:  12  12  12  10
```

This sample output shows that logical CPU ids are contiguous within nodes. For example, the cores on Node 0 are 0-23, which are contiguous. This convention is expected from most systems but is not guaranteed and could be different based on a given OEM platform. Depending on your operating system, you should confirm these mappings when using CPU affinity for important performance decisions.

Check daemons and services that could alter default behavior:

Each OS and platform provides various tools and services that attempt to optimize a given system. There are daemons and services intended to change or allocate CPU assignments and memory to leverage locality. This can sometimes result in unintended behavior where NUMA-assigned CPUs and memory may migrate after the initial application launch. For example, `numad` is a Linux system daemon that monitors NUMA topology and resource usage. It will attempt to locate processes for efficient NUMA locality and affinity by dynamically adjusting to changing system conditions. Enabling this feature may interfere with and override the initial launch commands of a deployment like `physcpubind`, `membind`, etc. If user is observing unexpected process threads and memory migrations, check if this daemon is enabled.

Check NUMA optimization policies:

Most platform attempt to allocate CPU and memory resources optimal NUMA leverage. Some OS and virtualization platform may have many available policy options. For example Linux automatic NUMA balancing moves tasks (threads or processes) closer to the memory they are accessing. Most applications and deployments benefit from being close to memory, meaning that this feature is often be enabled by default. NUMA balancing can have undesired effects. Further, a user's ability to bind the process CPUs and memory to different NUMA nodes may cause this setting to interfere with expected behavior. Execute the following command to disable NUMA balancing:

```
echo 0 > /proc/sys/kernel/numa_balancing
```

OS-level settings that are not available in BIOS:

An OEM platform may not provide BIOS options to change certain settings, but users may have been able to change them at the OS level. **AMD EPYC Core C6 (CC6) States** (alternately named C2 at the OS level) is one such example.

4th Gen AMD EPYC processors have C-States associated to cores and the Infinity Data fabric (DF). Disabling processor core C- States is highly discouraged. The system BIOS includes options to disable DF-C States for low latency and jitter-sensitive use cases. You can execute the following command to disable the Core C6 (CC6) state for the for all of the CPUs in a given system:

```
cpupower idle-set –d 2
```

You can also selectively disable Core C6. For a dual-socket system with 96-core processors, use 0-191 in the command to disable C2 for all 192 cores by executing the following command:

```
cpupower -c 0-191 idle-set -d 2.
```