



Arm[®] Neoverse[™] N3 Core

Telemetry Specification

Non-Confidential

Copyright © 2024 Arm Limited (or its affiliates).
All rights reserved.

Issue 01

109530_0100_01_en



Arm® Neoverse™ N3 Core Telemetry Specification

This document is Non-Confidential.

Copyright © 2024 Arm Limited (or its affiliates). All rights reserved.

This document is protected by copyright and other intellectual property rights.

Arm only permits use of this document if you have reviewed and accepted [Arm's Proprietary Notice](#) found at the end of this document.

This document (109530_0100_01_en) was issued on 2024-09-23. There might be a later issue at <https://developer.arm.com/documentation/109530>

See also: [Proprietary notice](#) | [Product and document information](#) | [Useful resources](#)

Start reading

If you prefer, you can skip to [the start of the content](#).

Intended audience

This specification is useful for engineers to collect and analyze Arm® Neoverse™ N3 Core telemetry data to gain insights about a system's performance. Architects and system designers can also use it for resource characterization and platform tuning.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on <https://support.developer.arm.com>.

To provide feedback on the document, fill the following survey: <https://developer.arm.com/documentation-feedback-survey>.

Contents

1. Overview of the Neoverse N3 core Telemetry methodology.....	6
1.1 Documentation and resources.....	7
2. Telemetry features of Neoverse N3 core.....	8
3. CPU performance analysis methodology.....	11
3.1 Stage 1: Topdown analysis.....	12
3.1.1 Topdown Level 1 metric group.....	14
3.1.2 Topdown Frontend metric group.....	16
3.1.3 Topdown Backend metric group.....	18
3.2 Stage 2: Microarchitecture exploration.....	20
3.3 Core memory traffic.....	25
3.4 Remote cache access.....	25
4. Neoverse N3 Telemetry cheat-sheets and lookup tables.....	26
4.1 Metrics cheat sheet for Neoverse N3.....	26
4.2 PMU events cheat sheet for Neoverse N3.....	28
4.3 Metrics lookup table for Neoverse N3.....	30
4.4 PMU events lookup table for Neoverse N3.....	34
5. Metrics by metric group in Neoverse N3.....	46
5.1 Topdown_L1 metrics for Neoverse N3.....	46
5.2 Topdown_Frontend metrics for Neoverse N3.....	49
5.3 Topdown_Backend metrics for Neoverse N3.....	53
5.4 Cycle_Accounting metrics for Neoverse N3.....	57
5.5 General metrics for Neoverse N3.....	58
5.6 MPKI metrics for Neoverse N3.....	59
5.7 Miss_Ratio metrics for Neoverse N3.....	64
5.8 SVE_Effectiveness metrics for Neoverse N3.....	70
5.9 FP_Arithmetic_Intensity metrics for Neoverse N3.....	72
5.10 FP_Precision_Mix metrics for Neoverse N3.....	73
5.11 Branch_Effectiveness metrics for Neoverse N3.....	75
5.12 ITLB_Effectiveness metrics for Neoverse N3.....	78

5.13 DTLB_Effectiveness metrics for Neoverse N3.....	81
5.14 L1I_Cache_Effectiveness metrics for Neoverse N3.....	85
5.15 L1D_Cache_Effectiveness metrics for Neoverse N3.....	86
5.16 L2_Cache_Effectiveness metrics for Neoverse N3.....	88
5.17 LL_Cache_Effectiveness metrics for Neoverse N3.....	89
5.18 Operation_Mix metrics for Neoverse N3.....	91
6. PMU events by functional group in Neoverse N3.....	96
6.1 Bus (BUS) events for Neoverse N3.....	97
6.2 Chain (CHAIN) events for Neoverse N3.....	98
6.3 Exception (EXCEPTION) events for Neoverse N3.....	99
6.4 L1D_Cache (L1D CACHE) events for Neoverse N3.....	103
6.5 L1I_Cache (L1I CACHE) events for Neoverse N3.....	107
6.6 L2_Cache (L2 CACHE) events for Neoverse N3.....	109
6.7 L3_Cache (L3 CACHE) events for Neoverse N3.....	115
6.8 LL_Cache (LL CACHE) events for Neoverse N3.....	117
6.9 Memory (MEMORY) events for Neoverse N3.....	119
6.10 Retired (RETIRED) events for Neoverse N3.....	123
6.11 SPE (SPE) events for Neoverse N3.....	131
6.12 Spec_Operation (SPEC OPERATION) events for Neoverse N3.....	134
6.13 FP_Operation (FP OPERATION) events for Neoverse N3.....	142
6.14 Stall (STALL) events for Neoverse N3.....	144
6.15 General (GENERAL) events for Neoverse N3.....	152
6.16 TLB (TLB) events for Neoverse N3.....	153
6.17 SVE (SVE) events for Neoverse N3.....	160
6.18 Non_PMU (NON_PMU) events for Neoverse N3.....	164
6.19 TRACE (TRACE) events for Neoverse N3.....	165
6.20 Events without a functional group for Neoverse N3.....	168
7. Supplemental performance debug PMU events.....	173
Proprietary notice.....	174
Product and document information.....	176
Product status.....	176
Revision history.....	176
Conventions.....	177

Useful resources..... 180

1. Overview of the Neoverse N3 core Telemetry methodology

The Arm® Neoverse N3 Core Telemetry Specification describes the Topdown methodology, derived metrics, and Performance Monitoring Unit (PMU) events supported by the Arm Neoverse N3 core, also known as the processor.



This specification is applicable to all releases of the product.

This specification implements the framework provided by the [Arm® CPU Telemetry Solution Topdown Methodology Specification](#), which is referred to as the Architecture Specification. The reader is expected to read this document in conjunction with the Architecture Specification.

Arm Telemetry framework

This specification outlines the telemetry features implemented for the Arm Neoverse N3 core and follows the Arm Telemetry framework for CPUs defined in the Architecture Specification.

The following list provides a brief description of the Telemetry framework:

Events

Hardware performance monitoring events implemented by the core that contain raw data read from the registers or memory buffers.

Metrics

Derived mathematical relationships between events that provide insight into the system behavior. They are developed to abstract hardware details of the events from consumers of the telemetry data.

Metric groups

Group of metrics that can be analyzed together to investigate a bottleneck scenario or a specific resource in a given system.

Methodology

Actionable guidance, such as Arm Topdown methodology, to explain how to consume the different metrics and events for a specific usage model. Decision tree with a group of metrics that can be analyzed hierarchically to investigate a bottleneck scenario or a specific resource in a given system.

Tool support for profiling and monitoring

This specification is also available in a machine-readable format (JSON) to be consumed by profiling and monitoring tools. The JSON schema implements the Arm Telemetry framework from the Architecture Specification.

The JSON for the core is published in the open source [Arm® Telemetry Solution GitLab repository](#).

1.1 Documentation and resources

Arm products include a set of documents.

The documentation and resources for Neoverse N3 consist of:

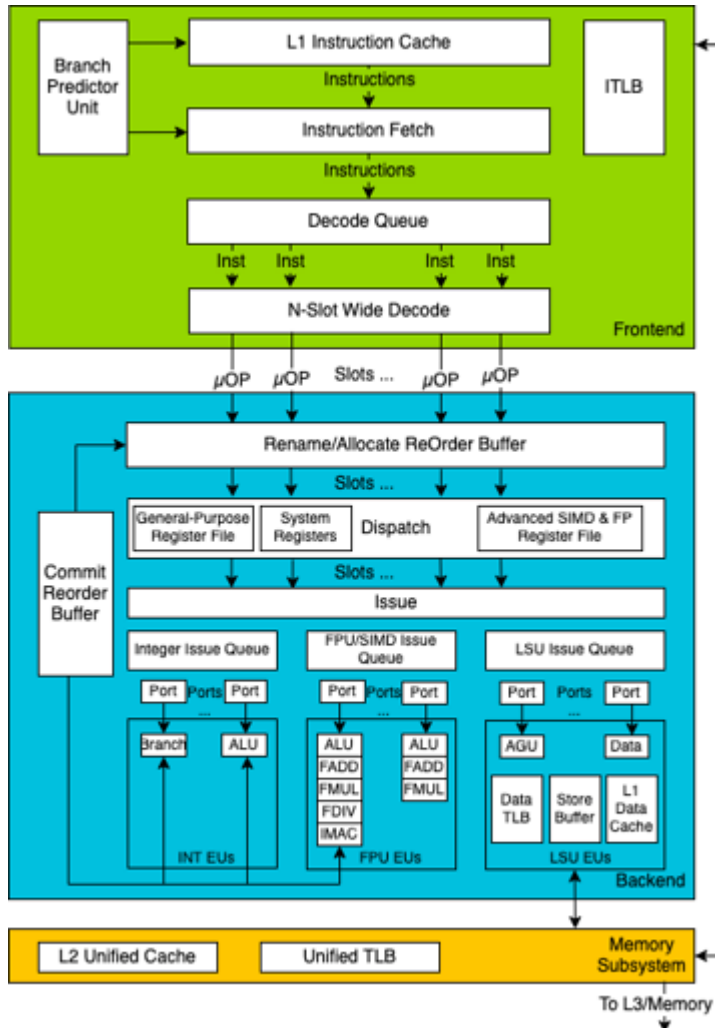
- [*Arm® Telemetry on Arm Developer*](#)
- [*Arm® Telemetry Solution GitLab repository*](#)
- [*Arm® Neoverse™ N3 Core Technical Reference Manual*](#)

2. Telemetry features of Neoverse N3 core

The Neoverse N3 core is a super pipelined superscalar processor that has an in-order frontend and out-of-order backend.

The following figure shows the microarchitecture details of the Neoverse N3 core.

Figure 2-1: Neoverse N3 core microarchitecture



The frontend of the core pipeline is comprised of the instruction fetch and decode units. The frontend also includes a branch predictor unit that predicts branch target addresses and fetches instructions ahead of the pipeline. This unit helps to hide latencies caused by control flow bubbles in the pipeline. The fetch unit can fetch multiple instructions for each cycle whose bandwidth is specific to a microarchitecture design, which gets stored in a decode queue. The decode queue sends multiple instructions per cycle for decoding, whose bandwidth is determined by the number of decode slots available. The decode unit decomposes the Arm architecture instructions into micro-operations, also known as micro-ops or (μ ops). This unit decodes more than one micro-

operation for each cycle, which are then fed to the rename unit for organization for out-of-order execution in the backend of the core. The instruction bandwidth is determined by the number of renamed slots available in the microarchitecture. The number of slots can be discovered in the SLOTS field of the PMMIR_EL1 registers. From a microarchitecture standpoint, the rename unit is considered the boundary between the frontend and backend of the core.

The backend of the core has a scheduler that orchestrates the operations to be executed when the issue queue associated with the operation can accept the operation. The issue queue sends operations for execution when the execution unit is free and the source operands are ready. Once the execution is complete, the results are sent to the commit Reorder Buffer (ROB) from where the instructions are retired when the speculated execution is confirmed. The backend of the core executes the operations out-of-order and stores results with the help of the reorder buffer. The dispatch unit tracks dependencies between operations and determines the operand availability for the execution of operations. Register renaming occurs at this stage to mitigate data dependency hazards.

In the dispatch unit, issue queues are employed for:

- Queuing the micro-operations (μ ops) to assigned ports
- Managing dependencies between operations
- Tracking operand availability for execution

Each execution port supports different categories of operations. After the execution of operations, the ROB is updated with execution results. Completed operations are retired architecturally in the right program order. Operations are flushed when the predicted program flow changes due to mispredictions or exceptions. The following execution units (EU) support typical operation types:

- Integer EU executes branches and arithmetic instructions.
- Floating-Point Unit (FPU)/Single Instruction Multiple Data (SIMD) EU executes the FP instruction and vector instructions.
- Load Store Unit executes load/store/atomic instructions.

The Memory subsystem of the core handles the execution of load and store operations which rely heavily on the memory hierarchy levels. The Neoverse N3 core has dedicated cache levels, L1 and L2 for each core, where the L2 cache is shared between the L1 data cache and the L1 instruction cache. The Load Store Unit controls the data flow between the caches and to memory. The core has multiple load/store units, which can handle both read and write operations. L1 and L2 caches are set-associative. The size of the cache is configurable for each implementation and determines the number of sets in each way. The private L2 cache of the core connects to the rest of the system through an AMBA® 5 CHI interface.

Neoverse N3 system configurations

All systems with the Arm® Neoverse™ Coherent Mesh Network support a shared system-level cache. Understanding the cache hierarchy and configuration of the system being analyzed is crucial in deriving insights from the cache effectiveness Performance Monitoring Unit (PMU) events.

It is always best to check with the Silicon Provider for details on the system configuration for the underlying system, including the cache sizes.

PMU capabilities of Neoverse N3

The Neoverse N3 core implements version 3.7 of the Performance Monitors Extension, `FEAT_PMUv3p7`, and Arm v8.4 debug architecture, `FEAT_Debugv8p4`.

For more information, see [Arm® Architecture Reference Manual for A-profile architecture](#).

The Neoverse N3 PMU has six configurable counter registers and one dedicated function counter to count CPU cycles.

3. CPU performance analysis methodology

The Arm Topdown methodology for the Neoverse N3 enables you to use PMU events, metrics, and metric groups to identify potential bottlenecks that could negatively impact the performance of the core in your design.

The methodology is conducted in two stages.

Stage 1: Topdown analysis

The first stage is to perform Topdown analysis to detect and identify any performance bottlenecks in the CPU, and provide direction for further analysis at Stage 2. Stage 1 uses hierarchical pipeline stall-related metrics. For more information, see [Stage 1: Topdown analysis](#).

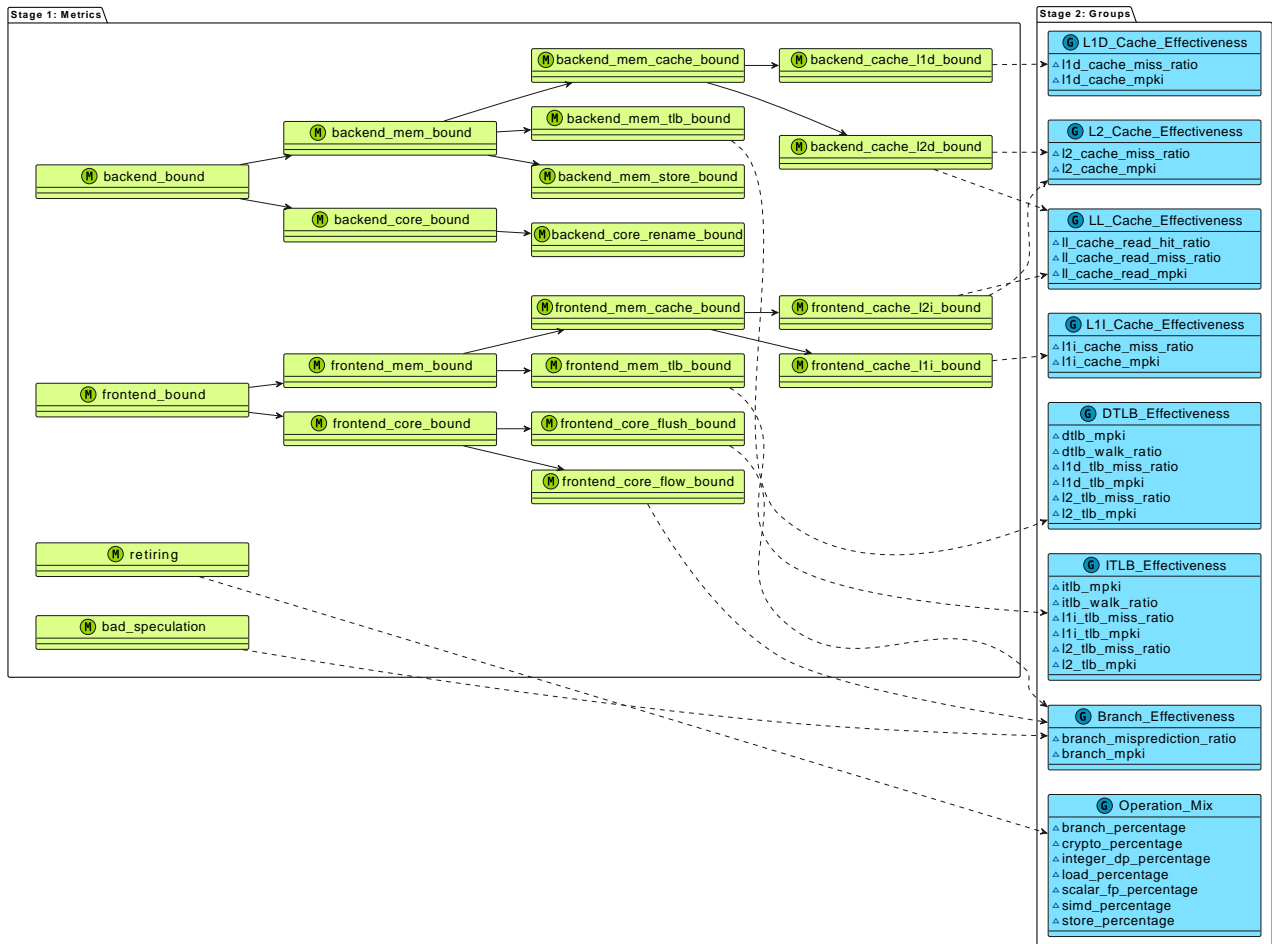
Stage 2: Microarchitecture Exploration

The second stage is to conduct microarchitecture exploration to further analyze bottlenecked CPU resources, based on the Stage 1 findings. Stage 2 uses a set of CPU resource-effectiveness metrics. For more information, see [Stage 2: Microarchitecture Exploration](#).

For more information about our approach to performance analysis and the standardized telemetry framework, see [Arm® CPU Telemetry Solution Topdown Methodology Specification](#).

We recommend collecting all metrics that are in Stage 1 and Stage 2 Topdown analysis for workload characterization. We provide a recommended set of microarchitecture exploration metric groups for further analysis, for hotspots detected in Stage 1. All Stage 2 metrics can be used to derive further insights into the overall microarchitecture behavior during the execution of the application under investigation. These metrics can be used independently of Stage 1.

The following figure gives an overview of the Topdown methodology tree for Neoverse N3. It shows Stage 1 metrics, and Stage 2 metric groups and metrics. Stage 1 covers the stall-related metrics for Topdown analysis for bottleneck identification. Stage 2 covers the microarchitecture exploration metric groups for root cause analysis.

Figure 3-1: Topdown methodology overview for Neoverse N3

The industry-standard metrics Misses Per Kilo Instructions (MPKI) and Miss Ratios are metric groups that are defined in Stage 2, but they are not included in the decision tree.

3.1 Stage 1: Topdown analysis

The objective of the Topdown analysis is to identify potential bottlenecks in the key blocks of the processor. Each block can be further broken down to units and subunits within the block that all have different performance characteristics. For example, control flow and data flow issues can be fixed differently in software after root cause analysis.

Results from a Topdown analysis can indicate:

- Which metric groups and metrics to analyze next.
- Areas where software improvements can be made in the current design.

- Existing microarchitectural limitations that can inform future hardware improvements, better system configuration, or tuning decisions at a platform level.

In Neoverse N3 there are three Stage 1 metric groups.

Topdown Level 1

The [Topdown Level 1 metric group](#) contains the first set of metrics to begin Topdown analysis of application performance. These metrics provide the percentage distribution of processor pipeline utilization.

For more information about the metrics in this group and the associated formulas and events, see [Topdown_L1](#).

Topdown Frontend

The [Topdown Frontend metric group](#) contains a set of metrics to analyze a frontend bound workload.

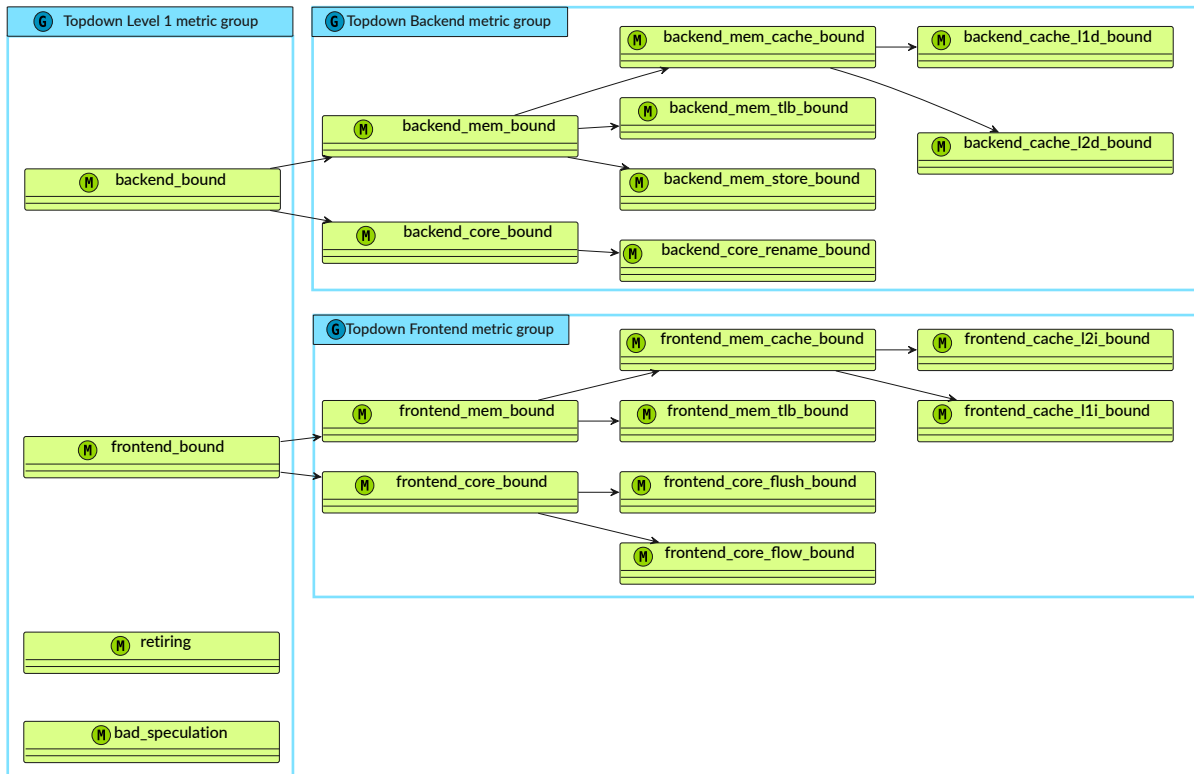
For more information about the metrics in this group and the associated formulas and events, see [Topdown_Frontend](#).

Topdown Backend

The [Topdown Backend metric group](#) contains a set of metrics to analyze a backend bound workload.

For more information about the metrics in this group and the associated formulas and events, see [Topdown_Backend](#).

The following figure shows the metric groups and metrics in the Stage 1 Topdown methodology tree for Neoverse N3, which supports up to four levels of hierarchical pipeline stall accounting.

Figure 3-2: Neoverse N3 Topdown methodology Stage 1 overview

3.1.1 Topdown Level 1 metric group

The [Topdown_L1](#) metric group contains the first set of metrics to begin Topdown analysis of application performance. These metrics provide the percentage distribution of processor pipeline utilization.

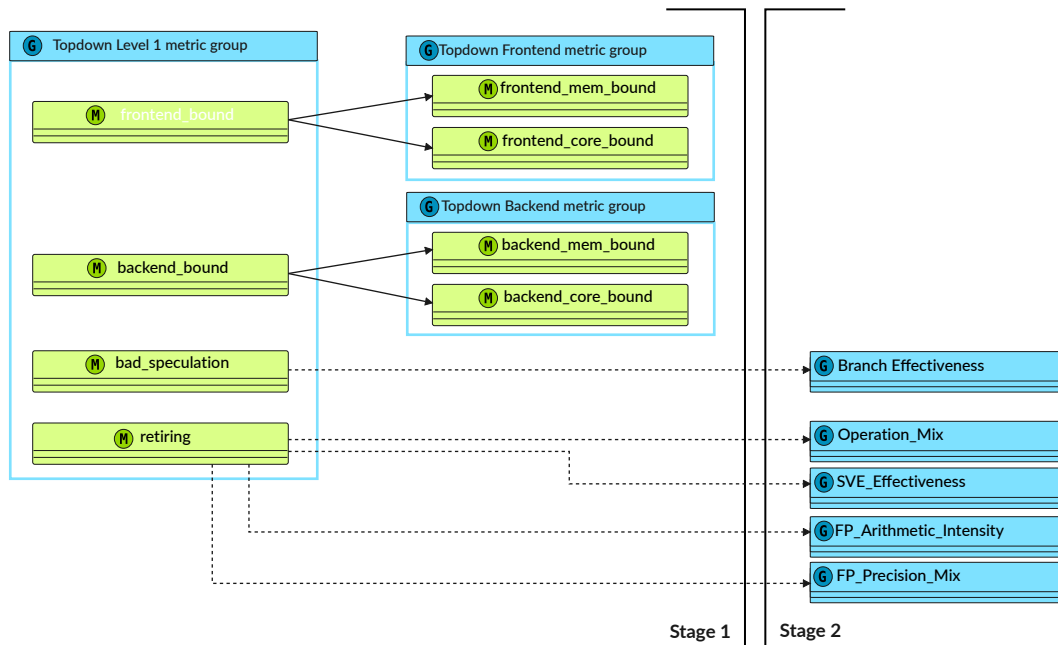
The following implementation criteria apply for this metric group:

- The sum of the metrics in Topdown_L1 equals 100% of the total execution cycles in the core implementation.
- The total execution bandwidth of the core is the same as the number of execution slots for operations.
- Slots are the number of ports in the rename unit, which is considered the boundary between the frontend and backend of the core.
- A frontend stall is counted when there are no micro-ops to rename.
- A backend stall is counted when there are micro-ops in the rename unit, but they cannot dispatch to the backend due to backend resource constraints.

- In a cycle where there are no micro-ops to rename and no micro-ops dispatched to the backend, the CPU counts a frontend stall.
- The definition of the backend stall is linked to the slot that does not dispatch although there are micro-ops available in the rename unit.

The following figure shows the metrics for Topdown_L1 and the next steps in the methodology.

Figure 3-3: Neoverse N3 Metric Group: Topdown_L1 and next steps



The metrics in the Topdown Level 1 metric group are:

frontend_bound

The **frontend_bound** metric measures pipeline inefficiency due to slots that were stalled because of resource constraints in the frontend.

For further analysis, the next step is the Stage 2 **Topdown Frontend metric group**.

backend_bound

The **backend_bound** metric measures pipeline inefficiency due to slots that were stalled because of resource constraints in the backend.

For further analysis, the next step is the Stage 2 **Topdown Backend metric group**.

bad_speculation

The **bad_speculation** metric is the percentage of total slots that executed operations and did not retire due to a pipeline flush, which indicates cycles that were utilized inefficiently. This metric measures pipeline inefficiency caused by flushes in the pipeline, due to branch

mispredictions or machine clears. For workloads that demonstrate high bad speculation rates, the next step is to understand the branch performance in the workload.

Some key reasons for pipeline inefficiency are the same as the reasons that are identified in the `frontend_core_flush_bound` metric, which is part of the `frontend_core_bound` metric within the [Topdown Frontend metric group](#).

For further analysis, the next step is the Stage 2 [Branch Effectiveness metric group](#).

retiring

The [retiring](#) metric is the percentage of total slots that retired operations, which indicates cycles that were utilized efficiently. This metric covers the total slots that were utilized efficiently by the processor. A high retirement rate can also mean that there is an opportunity for further performance optimization.

Scalar code that shows high retirement can be optimized by vectorization if there is data parallelism. In some cases, the code that is executed can be made redundant or optimized. When you analyze high retirement rates, evaluation of the instruction or operation mix is a recommended next step in the characterization of execution units that are heavily utilized.

For further analysis, the next steps are Stage 2 [Operation Mix metric group](#), Stage 2 [Floating Point Arithmetic Intensity metric group](#), Stage 2 [Floating Point Precision metric group](#), and Stage 2 [SVE Effectiveness metric group](#).

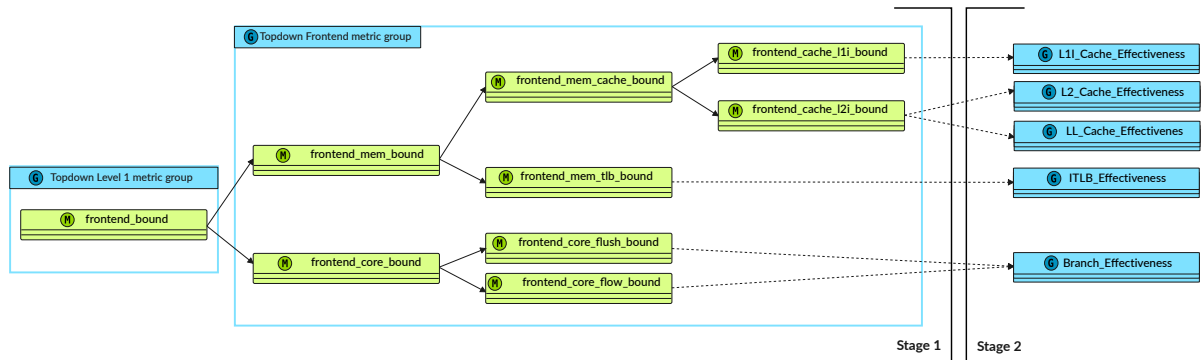
3.1.2 Topdown Frontend metric group

The [Topdown_Frontend](#) metric group contains a set of metrics to analyze a frontend bound workload.

The following implementation criteria apply for this metric group:

- The implementation of the Frontend Core Bound metric in Neoverse N3 is the difference between the Frontend Bound and Frontend Memory Bound metrics. This means that the cycle was stalled in the frontend and the stall was not caused by the frontend memory units. This derivation is mathematically correct but misses a direct count of the Frontend Core Bound metric from an implementation standpoint.
- Due to the implementation, the sum of the Frontend Core Bound and Frontend Memory Bound metrics is equal to 100% of the Frontend Bound metric.

The following figure shows the metrics for the Topdown_Frontend metric group and the next steps in the methodology.

Figure 3-4: Neoverse N3 Metric Group: Topdown_Frontend and next steps

The metrics in the Topdown Frontend metric group are:

frontend_mem_bound

The `frontend_mem_bound` metric is the percentage of total cycles stalled in the frontend due to frontend core resource constraints related to the instruction fetch latency issues caused by memory access components.

This metric breaks down into the following metrics:

- `frontend_mem_cache_bound`
 - `frontend_cache_l1i_bound`
 - `frontend_cache_l2i_bound`
- `frontend_mem_tlb_bound`

The following implementation criteria apply for these metrics:

- The Frontend Memory Cache Bound metric is implemented as the number of cycles of either the Frontend Memory L1I Bound metric or the Frontend Memory L2I Bound metric.
- The Frontend Memory L1I Bound metric counts when waiting on an L1 miss but not waiting on an L2 miss.

`frontend_mem_*_bound` stall metrics are derived ratios of Translation Lookaside Buffer (TLB) and Cache stalls such as:

- `frontend_mem_l1i_bound` is a ratio of frontend memory bound cycles waiting on an L1 miss.
- `frontend_mem_l2i_bound` is a ratio of frontend memory bound cycles waiting on an L2 miss.
- `frontend_mem_tlb_bound` is a ratio of frontend memory bound cycles waiting on a TLB miss.

For further analysis:

- The next step for `frontend_cache_l1i_bound` is the Stage 2 [Level 1 Instruction Cache Effectiveness metric group](#).
- The next step for `frontend_cache_l2i_bound` is the Stage 2 [Level 2 Unified Cache Effectiveness metric group](#) and the Stage 2 [Last Level Cache Effectiveness metric group](#).
- The next step for `frontend_mem_tlb_bound` is the Stage 2 [Instruction TLB Effectiveness metric group](#).

frontend_core_bound

The [frontend_core_bound](#) metric is the percentage of total cycles stalled in the frontend due to frontend core resource constraints not related to instruction fetch latency issues caused by memory access components.

Reasons for a stall in the frontend due to the core units are as follows:

- Resteer: Refers to the frontend resteer that cause PC updates, which can either be:
 - A flush that is caused by mispredictions
 - Other machine clears due to microarchitectural reasons such as exceptions and memory hazards
- Flow: Refers to the frontend structural flow stall that is caused in Fetch/Decode unit fetches due to a branch prediction unit that does not provide predictions on time.

This metric breaks down into the following metrics:

- [frontend_core_flush_bound](#)
- [frontend_core_flow_bound](#)

For further analysis, the next step is the Stage 2 [Branch Effectiveness metric group](#).



Some frontend core bound stall reasons are not captured by the `frontend_core_flush_bound` and `frontend_core_flow_bound` metrics.

3.1.3 Topdown Backend metric group

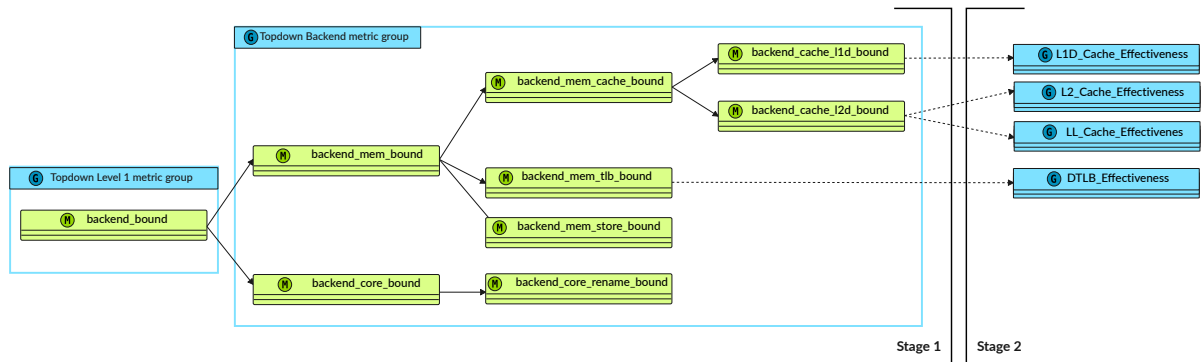
The [Topdown_Backend](#) metric group contains a set of metrics to analyze a backend bound workload.

The following implementation criteria apply for this metric group:

- The implementation of the Backend Core Bound metric in Neoverse N3 is the difference between the Backend Bound and Backend Memory Bound metrics. This means that the cycle was stalled in the backend and the stall was not caused by the backend memory units. This derivation is mathematically correct but misses a direct count of the Backend Core Bound metric from an implementation standpoint.
- Due to the implementation, the sum of the Backend Core Bound and Backend Memory Bound metrics is equal to 100% of the Backend Bound metric.

The following figure shows the metrics for the Topdown_Backend metric group and the next steps in the methodology.

Figure 3-5: Neoverse N3 Metric Group: Topdown_Backend and next steps



The metrics in the Topdown Backend metric group are:

backend_mem_bound

The `backend_mem_bound` metric is the percentage of total cycles stalled in the backend due to backend core resource constraints related to memory access latency issues caused by memory access components. `backend_mem_bound` counts the stall cycles in which no operation can be sent to the dispatch unit and a memory operation is being executed.

This metric breaks down into the following metrics:

- `backend_mem_cache_bound`
 - `backend_cache_l1d_bound`
 - `backend_cache_l2d_bound`
- `backend_mem_tlb_bound`
- `backend_mem_store_bound`

The following implementation criteria apply for these metrics:

- The Backend_Memory Cache Bound metric is implemented as the number of cycles of either the Backend Memory L1D Bound metric or the Backend Memory L1D Bound metric.
- The Backend Memory L1D Bound metric counts when waiting on an L1 miss but not waiting on an L2 miss.

`backend_mem_*_bound` stall metrics are derived ratios of TLB and Cache stalls such as:

- `backend_mem_l1d_bound` is a ratio of backend memory bound cycles waiting on an L1 data cache miss.
- `backend_mem_l2d_bound` is a ratio of backend memory bound cycles waiting on an L2 unified cache miss.

- `backend_mem_tlb_bound` is a ratio of backend memory bound cycles waiting on data for a TLB miss or walk.

The Backend Memory Store Bound metric also counts backend memory bound stall cycles while stores are outstanding.

For further analysis:

- The next step for `backend_cache_11d_bound` is the Stage 2 [Level 1 Data Cache Effectiveness metric group](#).
- The next step for `backend_cache_12d_bound` is the Stage 2 [Level 2 Unified Cache Effectiveness metric group](#) and the Stage 2 [Last Level Cache Effectiveness metric group](#).
- The next step for `backend_mem_tlb_bound` is the Stage 2 [Data TLB Effectiveness metric group](#).
- There is no further analysis for `backend_mem_store_bound`.

backend_core_bound

The [backend_core_bound](#) metric is the percentage of total cycles stalled in the backend due to backend core resource constraints not related to instruction fetch latency issues caused by memory access components.

Reasons for a stall in the backend due to the core units are as follows:

- **Rename:** Refers to rename stalls caused by the register renaming unit that sends operations to the dispatch unit.
- **Dispatch Busy:** Refers to execution stalls caused by either issue queue unit stalls, execution port starvation, or data dependency issues for operations to make further progress in execution.

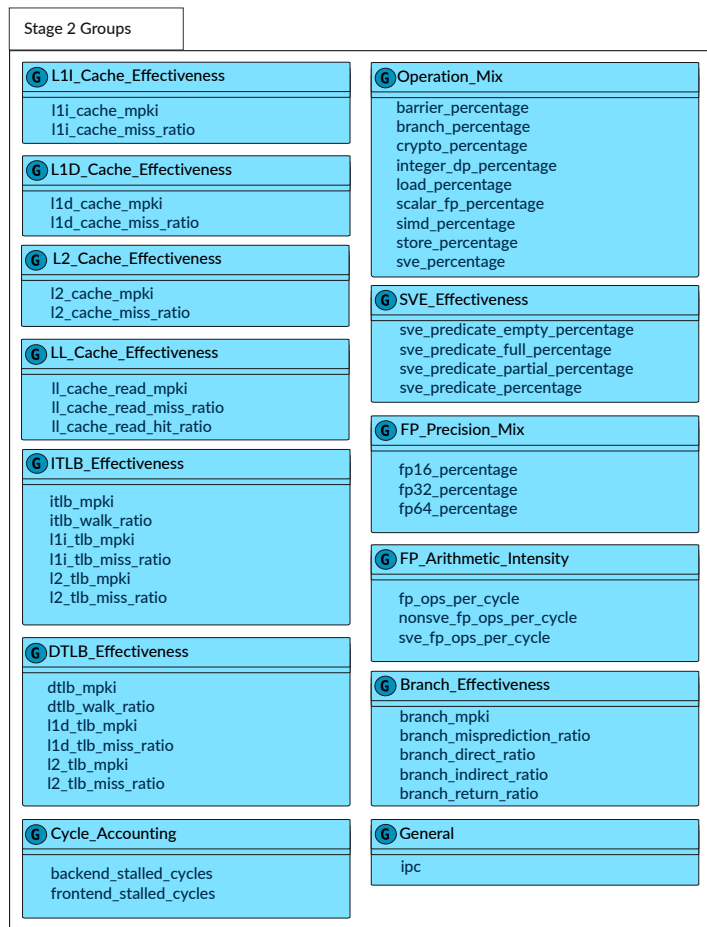
For further analysis, the next step is the Stage 1 [backend_core_rename_bound](#) metric.

3.2 Stage 2: Microarchitecture exploration

Topdown analysis Stage 2 helps to locate the hot spots in the code and conduct root cause analysis. This stage reveals more details about the CPU component that caused the bottleneck and was identified during the Stage 1 process.

Stage 2 contains metric groups for microarchitecture exploration targeted at CPU resource effectiveness and other relevant metrics. Metrics in this category are designed for users who are interested in the microarchitectural characteristics of key CPU components.

The following figure shows the Stage 2 metric groups for Neoverse N3.

Figure 3-6: Neoverse N3 Topdown methodology Stage 2 overview

In Neoverse N3 there are two metric groups based on industry standard metrics, Misses Per Kilo Instructions (MPKI) and Miss Ratios, plus eight other Stage 2 metric groups.

Level 1 Data Cache Effectiveness metric group

The [L1D_Cache_Effectiveness](#) metric group contains metrics to evaluate the effectiveness of the L1 Data Cache on the processor.

Level 1 Instruction Cache Effectiveness metric group

The [L1I_Cache_Effectiveness](#) metric group contains metrics to evaluate the effectiveness of the L1 Instruction Cache on the processor.

Level 2 Unified Cache Effectiveness metric group

The [L2_Cache_Effectiveness](#) metric group contains metrics to evaluate the effectiveness of the L2 Unified Cache on the processor.

Last Level Cache Effectiveness metric group

The [LL_Cache_Effectiveness](#) metric group contains metrics to evaluate the effectiveness of the Last Level Cache on the processor.

In systems that support a shared System Level Cache (SLC) in the interconnect that is configured to count Last Level Cache events:

- The [Last level cache access, read](#) event counts total SLC accesses made by the core.
- The [Last level cache miss, read](#) event counts accesses missed at the SLC.

The [ll_cache_read_mpki](#) and [ll_cache_read_miss_ratio](#) metrics can be used to analyze the last level read behavior.

Another useful metric to measure the SLC hit percentage for read traffic is [ll_cache_read_hit_ratio](#).

Last level cache events do not have a write variant in Neoverse N3 because the SLC is only used as an eviction cache for the core. In addition, all the writes complete early at the interconnect when the transaction is acknowledged but always completed.

Data TLB Effectiveness and Instruction TLB Effectiveness metric groups

The [DTLB_Effectiveness](#) and [ITLB_Effectiveness](#) metric groups contain metrics to evaluate the effectiveness of the data TLB and instruction TLB, respectively, on the processor.

An important performance evaluation step is to check the virtual memory system performance, which affects the instruction fetch performance in the frontend and memory access performance on the data side.

The processor translates a virtual address into a physical address for any instruction or data memory access before it accesses the respective cache.



A program's view of memory is the virtual address, but the processor works with the physical address when it accesses cache or memory.

Virtual to physical mappings are defined in the page translation tables which reside in system memory. Access to these tables requires one or more memory operations that take many cycles to complete and are known as a translation table walk. However, TLBs cache these translation table walks, which significantly reduces the number of accesses to system memory and makes translations faster.

Several key metrics can be used to evaluate the TLB effectiveness and the cost of latency that is specifically caused by translation table walks:

- The [itlb_mpki](#) and [dtlb_mpki](#) metrics provide the rate of TLB Walks per kilo instructions for instruction and data accesses, respectively. These metrics help to evaluate and correlate the TLB efficiency with respect to the total number of instructions.

- The [dtlb_walk_ratio](#) metric provides the ratio of DTLB Walks to the overall TLB lookups made by the program. this metric is the same as the [DTLB_WALK](#) and [MEM_ACCESS](#) events because every MEM_ACCESS causes a [L1D_TLB](#) access.
- The [itlb_walk_ratio](#) metric provides a percentage of ITLB walks to the overall TLB lookups initiated from the instruction side.

Branch Effectiveness metric group

The [Branch_Effectiveness](#) metric group contains metrics to evaluate the effectiveness of branch instruction execution on the processor.

Branch mispredictions are costly in a deeply pipelined CPU, causing pipeline flushes and wasted cycles. Workloads contain, on average, one branch in every six instructions. Although modern CPUs have optimized branch prediction units, there are many use cases that are branch heavy and hard to predict. Examples of such use cases include ray tracing and decision tree algorithms. In some of these cases, there can be hundreds of unique branch paths to take and the target might be input data dependent.

Two key performance metrics can be used for a high-level evaluation of the branch execution performance regarding the overall program execution:

- The [branch_mpki](#) metric provides total branch mispredictions per kilo instructions.
- The [branch_misprediction_ratio](#) metric can indicate the ratio of branches that were mispredicted to the overall branches.

Branch prediction units work differently depending on the branch type. The main components are:

Branch History Table (BHT)

Stores the history of conditional branches, taken or not.

Branch Target Buffer (BTB)

Stores the target address for indirect branches.

Return Address Stack (RAS)

Stores the function return branches.

Getting a breakdown of the branch type helps to investigate each subunit within the branch prediction unit.

Cycle Accounting metric group

The [Cycle_Accounting](#) metric group contains a set of metrics that measure the percentage of processor cycles stalled in either frontend or backend of the processor.

General metric group

The [General](#) metric group contains general CPU metrics for performance analysis such as Instructions Per Cycle (IPC).

Floating Point Arithmetic Intensity metric group

The [FP_Arithmetic_Intensity](#) metric group provides metrics to evaluate the effectiveness of floating point operation execution on this processor.

The metrics evaluate operations by computation and vector lanes, metrics are provided for scalable vector operations and non scalable vector operations.

Floating Point Precision metric group

The [FP_Precision_Mix](#) metric group provides metrics to evaluate tmix of precision of floating point instruction execution on this processor.

SVE Effectiveness metric group

The [SVE_Effectiveness](#) metric group provides metrics to evaluate the effectiveness of predicated SVE instruction execution on this processor.

Operation Mix metric group

The [Operation_Mix](#) metric group provides the distribution of micro-operation types executed for the program.

The Neoverse N3 microarchitecture has a variety of execution units that can process more than six types of operations:

- Branch
- Single-cycle integers
- Multicycle integers
- Load/store unit with address generation
- Advanced floating-point/SIMD operations
- SVE

The operations that are issued to these execution units can be counted by the PMU events in the Operation Mix metric group.

The following implementation criteria apply for this metric group:

- The [sve_all_percentage](#) operation includes SVE load and store operations. These operations are also counted in the [load_percentage](#) and [store_percentage](#) metrics. Therefore, this duplicate counting should be taken into account when these metrics are considered together.

Misses Per Kilo Instructions metric group

The [MPKI](#) metric group contains metrics for different CPU resources that can be measured as misses per kilo instructions. These metrics can be used to normalize the misses in CPU components against the total instructions executed. The primary components are branches, caches, and TLBs.

MPKI is an industry-standard metric that can also help with comparisons across different implementations of the Arm architecture, because instructions retired should count the same on all AArch64-based microarchitectures.



The MPKI metric group is a Stage 2 metric but is not included in the methodology decision tree.

Miss Ratio metric group

The [Miss_Ratio](#) metric group contains metrics to measure miss ratios of different processor resources. These metrics can be used to calculate the ratio of misses in CPU components against the total accesses in those components. The primary components are branches, caches, and TLBs.

Miss Ratio metrics provide insights into the efficiency of each CPU component in the pipeline and help to find the root cause of issues.



The Miss_Ratio metric group is a Stage 2 metric but is not included in the methodology decision tree.

3.3 Core memory traffic

The `MEM_ACCESS` event counts the total number of memory operations that were issued by the Load Store Unit (LSU) of the core.

Because these operations are looked up in the `L1D_CACHE` first, both the `L1D_CACHE` and `MEM_ACCESS` events count at the same rate.

Neoverse N3 also supports two additional events, [MEM_ACCESS_RD](#) and [MEM_ACCESS_WR](#) that can provide the read and write traffic breakdown respectively. These events are not the same as the [LD_SPEC](#) and [ST_SPEC](#) events because they count memory operations that are speculatively issued but not always executed.

3.4 Remote cache access

For systems with multiple sockets or SoCs, Neoverse N3 supports the `REMOTE_ACCESS` event, which counts the memory transactions that were completed by a subordinate source from another chip.

4. Neoverse N3 Telemetry cheat-sheets and lookup tables

The cheat-sheets and lookup tables enable you to find and access metrics and events in different ways.

Cheat-sheets

Both metrics and events are listed by metric groups.

Lookup tables

Metrics are listed alphabetically, with the related events, and metric groups.

Events are listed by code number, with the related metrics, metric groups, and functional groups.

4.1 Metrics cheat sheet for Neoverse N3

Metrics are listed in their respective metric groups. Some metrics are used in more than one metric group.

Neoverse N3 specification provides the following types of metrics:

- Total implemented Common metrics: 67

Topdown Level 1 (4)	Topdown Frontend (8)	Topdown Backend (9)
<ul style="list-style-type: none"> • backend_bound • bad_speculation • frontend_bound • retiring 	<ul style="list-style-type: none"> • frontend_cache_l1i_bound • frontend_cache_l2i_bound • frontend_core_bound • frontend_core_flow_bound • frontend_core_flush_bound • frontend_mem_bound • frontend_mem_cache_bound • frontend_mem_tlb_bound 	<ul style="list-style-type: none"> • backend_busy_bound • backend_cache_l1d_bound • backend_cache_l2d_bound • backend_core_bound • backend_core_rename_bound • backend_mem_bound • backend_mem_cache_bound • backend_mem_store_bound • backend_mem_tlb_bound

Cycle Accounting (2)	General (1)	Misses Per Kilo Instructions (10)
<ul style="list-style-type: none"> backend_stalled_cycles frontend_stalled_cycles 	<ul style="list-style-type: none"> ipc 	<ul style="list-style-type: none"> branch_mpki dtlb_mpki itlb_mpki l1d_cache_mpki l1d_tlb_mpki l1i_cache_mpki l1i_tlb_mpki l2_cache_mpki l2_tlb_mpki ll_cache_read_mpki
Miss Ratio (10)	SVE Effectiveness (4)	Floating Point Arithmetic Intensity (3)
<ul style="list-style-type: none"> branch_misprediction_ratio dtlb_walk_ratio itlb_walk_ratio l1d_cache_miss_ratio l1d_tlb_miss_ratio l1i_cache_miss_ratio l1i_tlb_miss_ratio l2_cache_miss_ratio l2_tlb_miss_ratio ll_cache_read_miss_ratio 	<ul style="list-style-type: none"> sve_predicate_empty_percentage sve_predicate_full_percentage sve_predicate_partial_percentage sve_predicate_percentage 	<ul style="list-style-type: none"> fp_ops_per_cycle nonsve_fp_ops_per_cycle sve_fp_ops_per_cycle
Floating Point Precision (3)	Branch Effectiveness (5)	Instruction TLB Effectiveness (6)
<ul style="list-style-type: none"> fp16_percentage fp32_percentage fp64_percentage 	<ul style="list-style-type: none"> branch_direct_ratio branch_indirect_ratio branch_misprediction_ratio branch_mpki branch_return_ratio 	<ul style="list-style-type: none"> itlb_mpki itlb_walk_ratio l1i_tlb_miss_ratio l1i_tlb_mpki l2_tlb_miss_ratio l2_tlb_mpki
Data TLB Effectiveness (6)	L1 Instruction Cache Effectiveness (2)	L1 Data Cache Effectiveness (2)
<ul style="list-style-type: none"> dtlb_mpki dtlb_walk_ratio l1d_tlb_miss_ratio l1d_tlb_mpki l2_tlb_miss_ratio l2_tlb_mpki 	<ul style="list-style-type: none"> l1i_cache_miss_ratio l1i_cache_mpki 	<ul style="list-style-type: none"> l1d_cache_miss_ratio l1d_cache_mpki

L2 Unified Cache Effectiveness (2)	Last Level Cache Effectiveness (3)	Speculative Operation Mix (9)
<ul style="list-style-type: none"> l2_cache_miss_ratio l2_cache_mpki 	<ul style="list-style-type: none"> ll_cache_read_hit_ratio ll_cache_read_miss_ratio ll_cache_read_mpki 	<ul style="list-style-type: none"> barrier_percentage branch_percentage crypto_percentage integer_dp_percentage load_percentage scalar_fp_percentage simd_percentage store_percentage sve_all_percentage

4.2 PMU events cheat sheet for Neoverse N3

Events are listed in their respective metric groups. Some events are not used in the Methodology, therefore are not shown in the cheat sheet.

Neoverse N3 specification provides the following types of PMU events:

- Total implemented Common events: 202
- Total Implemented Product ImpDef events: 12
- PMU Only events : 12
- ETE Only events : 2

Topdown Level 1 (7)	Topdown Frontend (8)	Topdown Backend (9)
<ul style="list-style-type: none"> CPU_CYCLES OP_RETIRED OP_SPEC STALL_FRONTEND_FLUSH STALL_SLOT STALL_SLOT_BACKEND STALL_SLOT_FRONTEND 	<ul style="list-style-type: none"> STALL_FRONTEND STALL_FRONTEND_CPUBOUND STALL_FRONTEND_FLOW STALL_FRONTEND_FLUSH STALL_FRONTEND_L1I STALL_FRONTEND_MEM STALL_FRONTEND_MEMBOUND STALL_FRONTEND_TLB 	<ul style="list-style-type: none"> STALL_BACKEND STALL_BACKEND_BUSY STALL_BACKEND_CPUBOUND STALL_BACKEND_L1D STALL_BACKEND_MEM STALL_BACKEND_MEMBOUND STALL_BACKEND_RENAME STALL_BACKEND_ST STALL_BACKEND_TLB

Cycle Accounting (3)	General (2)	Misses Per Kilo Instructions (11)
<ul style="list-style-type: none"> CPU_CYCLES STALL_BACKEND STALL_FRONTEND 	<ul style="list-style-type: none"> CPU_CYCLES INST_RETIRED 	<ul style="list-style-type: none"> BR_MIS_PRED_RETIRED DTLB_WALK INST_RETIRED ITLB_WALK L1D_CACHE_REFILL L1D_TLB_REFILL L1I_CACHE_REFILL L1I_TLB_REFILL L2D_CACHE_REFILL L2D_TLB_REFILL LL_CACHE_MISS_RD
Miss Ratio (18)	SVE Effectiveness (5)	Floating Point Arithmetic Intensity (2)
<ul style="list-style-type: none"> BR_MIS_PRED_RETIRED BR_RETIRED DTLB_WALK ITLB_WALK L1D_CACHE L1D_CACHE_REFILL L1D_TLB L1D_TLB_REFILL L1I_CACHE L1I_CACHE_REFILL L1I_TLB L1I_TLB_REFILL L2D_CACHE L2D_CACHE_REFILL L2D_TLB L2D_TLB_REFILL LL_CACHE_MISS_RD LL_CACHE_RD 	<ul style="list-style-type: none"> INST_SPEC SVE_PRED_EMPTY_SPEC SVE_PRED_FULL_SPEC SVE_PRED_PARTIAL_SPEC SVE_PRED_SPEC 	<ul style="list-style-type: none"> FP_FIXED_OPS_SPEC FP_SCALE_OPS_SPEC
Floating Point Precision (4)	Branch Effectiveness (6)	Instruction TLB Effectiveness (6)
<ul style="list-style-type: none"> FP_DP_SPEC FP_HP_SPEC FP_SP_SPEC INST_SPEC 	<ul style="list-style-type: none"> BR_IMMED_RETIRED BR_IND_RETIRED BR_MIS_PRED_RETIRED BR_RETIRED BR_RETURN_RETIRED INST_RETIRED 	<ul style="list-style-type: none"> INST_RETIRED ITLB_WALK L1I_TLB L1I_TLB_REFILL L2D_TLB L2D_TLB_REFILL

Data TLB Effectiveness (6)	L1 Instruction Cache Effectiveness (3)	L1 Data Cache Effectiveness (3)
<ul style="list-style-type: none"> DTLB_WALK INST_RETIRED L1D_TLB L1D_TLB_REFILL L2D_TLB L2D_TLB_REFILL 	<ul style="list-style-type: none"> INST_RETIRED L1I_CACHE L1I_CACHE_REFILL 	<ul style="list-style-type: none"> INST_RETIRED L1D_CACHE L1D_CACHE_REFILL
L2 Unified Cache Effectiveness (3)	Last Level Cache Effectiveness (3)	Speculative Operation Mix (12)
<ul style="list-style-type: none"> INST_RETIRED L2D_CACHE L2D_CACHE_REFILL 	<ul style="list-style-type: none"> INST_RETIRED LL_CACHE_MISS_RD LL_CACHE_RD 	<ul style="list-style-type: none"> ASE_SPEC CRYPTO_SPEC DMB_SPEC DP_SPEC DSB_SPEC INST_SPEC ISB_SPEC LD_SPEC PC_WRITE_SPEC ST_SPEC SVE_INST_SPEC VFP_SPEC

4.3 Metrics lookup table for Neoverse N3

All metrics are listed alphabetically, with the related events, and metric groups. Some metrics are used in more than one metric group, in that case they are listed multiple times so that you can jump to the most relevant metric group for your requirements.

Table 4-13: Metrics listed by name, with related events and metric groups

Metric Name	Formula from Events	Metric Groups
backend_bound	$\text{STALL_SLOT_BACKEND} / (5 * \text{CPU_CYCLES}) * 100$	<ul style="list-style-type: none"> Topdown_L1
backend_busy_bound	$\text{STALL_BACKEND_BUSY} / \text{STALL_BACKEND} * 100$	<ul style="list-style-type: none"> Topdown_Backend
backend_cache_l1d_bound	$\text{STALL_BACKEND_L1D} / (\text{STALL_BACKEND_L1D} + \text{STALL_BACKEND_MEM}) * 100$	<ul style="list-style-type: none"> Topdown_Backend
backend_cache_l2d_bound	$\text{STALL_BACKEND_MEM} / (\text{STALL_BACKEND_L1D} + \text{STALL_BACKEND_MEM}) * 100$	<ul style="list-style-type: none"> Topdown_Backend
backend_core_bound	$\text{STALL_BACKEND_CPUBOUND} / \text{STALL_BACKEND} * 100$	<ul style="list-style-type: none"> Topdown_Backend

Metric Name	Formula from Events	Metric Groups
backend_core_rename_bound	$\text{STALL_BACKEND_RENAME} / \text{STALL_BACKEND_CPUBOUND} * 100$	<ul style="list-style-type: none"> Topdown_Backend
backend_mem_bound	$\text{STALL_BACKEND_MEMBOUND} / \text{STALL_BACKEND} * 100$	<ul style="list-style-type: none"> Topdown_Backend
backend_mem_cache_bound	$(\text{STALL_BACKEND_L1D} + \text{STALL_BACKEND_MEM}) / \text{STALL_BACKEND_MEMBOUND} * 100$	<ul style="list-style-type: none"> Topdown_Backend
backend_mem_store_bound	$\text{STALL_BACKEND_ST} / \text{STALL_BACKEND_MEMBOUND} * 100$	<ul style="list-style-type: none"> Topdown_Backend
backend_mem_tlb_bound	$\text{STALL_BACKEND_TLB} / \text{STALL_BACKEND_MEMBOUND} * 100$	<ul style="list-style-type: none"> Topdown_Backend
backend_stalled_cycles	$\text{STALL_BACKEND} / \text{CPU_CYCLES} * 100$	<ul style="list-style-type: none"> Cycle_Accounting
bad_speculation	$(1 - \text{STALL_SLOT} / (5 * \text{CPU_CYCLES})) * (1 - \text{OP_RETIRED} / \text{OP_SPEC}) * 100 + \text{STALL_FRONTEND_FLUSH} / \text{CPU_CYCLES} * 100$	<ul style="list-style-type: none"> Topdown_L1
barrier_percentage	$(\text{ISB_SPEC} + \text{DSB_SPEC} + \text{DMB_SPEC}) / \text{INST_SPEC} * 100$	<ul style="list-style-type: none"> Operation_Mix
branch_direct_ratio	$\text{BR_IMMED_RETIRED} / \text{BR_RETIRED}$	<ul style="list-style-type: none"> Branch_Effectiveness
branch_indirect_ratio	$\text{BR_IND_RETIRED} / \text{BR_RETIRED}$	<ul style="list-style-type: none"> Branch_Effectiveness
<ul style="list-style-type: none"> branch_misprediction_ratio in Branch_Effectiveness branch_misprediction_ratio in Miss_Ratio 	$\text{BR_MIS_PRED_RETIRED} / \text{BR_RETIRED}$	<ul style="list-style-type: none"> Branch_Effectiveness Miss_Ratio
<ul style="list-style-type: none"> branch_mpki in Branch_Effectiveness branch_mpki in MPKI 	$\text{BR_MIS_PRED_RETIRED} / \text{INST_RETIRED} * 1000$	<ul style="list-style-type: none"> Branch_Effectiveness MPKI
branch_percentage	$\text{PC_WRITE_SPEC} / \text{INST_SPEC} * 100$	<ul style="list-style-type: none"> Operation_Mix
branch_return_ratio	$\text{BR_RETURN_RETIRED} / \text{BR_RETIRED}$	<ul style="list-style-type: none"> Branch_Effectiveness
crypto_percentage	$\text{CRYPTO_SPEC} / \text{INST_SPEC} * 100$	<ul style="list-style-type: none"> Operation_Mix
<ul style="list-style-type: none"> dtlb_mpki in DTLB_Effectiveness dtlb_mpki in MPKI 	$\text{DTLB_WALK} / \text{INST_RETIRED} * 1000$	<ul style="list-style-type: none"> DTLB_Effectiveness MPKI
<ul style="list-style-type: none"> dtlb_walk_ratio in DTLB_Effectiveness dtlb_walk_ratio in Miss_Ratio 	$\text{DTLB_WALK} / \text{L1D_TLB}$	<ul style="list-style-type: none"> DTLB_Effectiveness Miss_Ratio
fp16_percentage	$\text{FP_HP_SPEC} / \text{INST_SPEC} * 100$	<ul style="list-style-type: none"> FP_Precision_Mix
fp32_percentage	$\text{FP_SP_SPEC} / \text{INST_SPEC} * 100$	<ul style="list-style-type: none"> FP_Precision_Mix
fp64_percentage	$\text{FP_DP_SPEC} / \text{INST_SPEC} * 100$	<ul style="list-style-type: none"> FP_Precision_Mix
fp_ops_per_cycle	$(\text{FP_SCALE_OPS_SPEC} + \text{FP_FIXED_OPS_SPEC}) / \text{CPU_CYCLES}$	<ul style="list-style-type: none"> FP_Arithmetic_Intensity
frontend_bound	$(\text{STALL_SLOT_FRONTEND} / (5 * \text{CPU_CYCLES}) - \text{STALL_FRONTEND_FLUSH} / \text{CPU_CYCLES}) * 100$	<ul style="list-style-type: none"> Topdown_L1

Metric Name	Formula from Events	Metric Groups
frontend_cache_l1i_bound	$\text{STALL_FRONTEND_L1I} / (\text{STALL_FRONTEND_L1I} + \text{STALL_FRONTEND_MEM}) * 100$	<ul style="list-style-type: none"> Topdown_Frontend
frontend_cache_l2i_bound	$\text{STALL_FRONTEND_MEM} / (\text{STALL_FRONTEND_L1I} + \text{STALL_FRONTEND_MEM}) * 100$	<ul style="list-style-type: none"> Topdown_Frontend
frontend_core_bound	$\text{STALL_FRONTEND_CPUBOUND} / \text{STALL_FRONTEND} * 100$	<ul style="list-style-type: none"> Topdown_Frontend
frontend_core_flow_bound	$\text{STALL_FRONTEND_FLOW} / \text{STALL_FRONTEND_CPUBOUND} * 100$	<ul style="list-style-type: none"> Topdown_Frontend
frontend_core_flush_bound	$\text{STALL_FRONTEND_FLUSH} / \text{STALL_FRONTEND_CPUBOUND} * 100$	<ul style="list-style-type: none"> Topdown_Frontend
frontend_mem_bound	$\text{STALL_FRONTEND_MEMBOUND} / \text{STALL_FRONTEND} * 100$	<ul style="list-style-type: none"> Topdown_Frontend
frontend_mem_cache_bound	$(\text{STALL_FRONTEND_L1I} + \text{STALL_FRONTEND_MEM}) / \text{STALL_FRONTEND_MEMBOUND} * 100$	<ul style="list-style-type: none"> Topdown_Frontend
frontend_mem_tlb_bound	$\text{STALL_FRONTEND_TLB} / \text{STALL_FRONTEND_MEMBOUND} * 100$	<ul style="list-style-type: none"> Topdown_Frontend
frontend_stalled_cycles	$\text{STALL_FRONTEND} / \text{CPU_CYCLES} * 100$	<ul style="list-style-type: none"> Cycle_Accounting
integer_dp_percentage	$(\text{DP_SPEC} - \text{DSB_SPEC}) / \text{INST_SPEC} * 100$	<ul style="list-style-type: none"> Operation_Mix
ipc	$\text{INST_RETIRED} / \text{CPU_CYCLES}$	<ul style="list-style-type: none"> General
<ul style="list-style-type: none"> itlb_mpki in ITLB_Effectiveness itlb_mpki in MPKI 	$\text{ITLB_WALK} / \text{INST_RETIRED} * 1000$	<ul style="list-style-type: none"> ITLB_Effectiveness MPKI
<ul style="list-style-type: none"> itlb_walk_ratio in ITLB_Effectiveness itlb_walk_ratio in Miss_Ratio 	$\text{ITLB_WALK} / \text{L1I_TLB}$	<ul style="list-style-type: none"> ITLB_Effectiveness Miss_Ratio
<ul style="list-style-type: none"> l1d_cache_miss_ratio in L1D_Cache_Effectiveness l1d_cache_miss_ratio in Miss_Ratio 	$\text{L1D_CACHE_REFILL} / \text{L1D_CACHE}$	<ul style="list-style-type: none"> L1D_Cache_Effectiveness Miss_Ratio
<ul style="list-style-type: none"> l1d_cache_mpki in L1D_Cache_Effectiveness l1d_cache_mpki in MPKI 	$\text{L1D_CACHE_REFILL} / \text{INST_RETIRED} * 1000$	<ul style="list-style-type: none"> L1D_Cache_Effectiveness MPKI
<ul style="list-style-type: none"> l1d_tlb_miss_ratio in DTLB_Effectiveness l1d_tlb_miss_ratio in Miss_Ratio 	$\text{L1D_TLB_REFILL} / \text{L1D_TLB}$	<ul style="list-style-type: none"> DTLB_Effectiveness Miss_Ratio
<ul style="list-style-type: none"> l1d_tlb_mpki in DTLB_Effectiveness l1d_tlb_mpki in MPKI 	$\text{L1D_TLB_REFILL} / \text{INST_RETIRED} * 1000$	<ul style="list-style-type: none"> DTLB_Effectiveness MPKI
<ul style="list-style-type: none"> l1i_cache_miss_ratio in L1I_Cache_Effectiveness l1i_cache_miss_ratio in Miss_Ratio 	$\text{L1I_CACHE_REFILL} / \text{L1I_CACHE}$	<ul style="list-style-type: none"> L1I_Cache_Effectiveness Miss_Ratio

Metric Name	Formula from Events	Metric Groups
<ul style="list-style-type: none"> l1i_cache_mpki in L1I_Cache_Effectiveness l1i_cache_mpki in MPKI 	$L1I_CACHE_REFILL / INST_RETIRED * 1000$	<ul style="list-style-type: none"> L1I_Cache_Effectiveness MPKI
<ul style="list-style-type: none"> l1i_tlb_miss_ratio in ITLB_Effectiveness l1i_tlb_miss_ratio in Miss_Ratio 	$L1I_TLB_REFILL / L1I_TLB$	<ul style="list-style-type: none"> ITLB_Effectiveness Miss_Ratio
<ul style="list-style-type: none"> l1i_tlb_mpki in ITLB_Effectiveness l1i_tlb_mpki in MPKI 	$L1I_TLB_REFILL / INST_RETIRED * 1000$	<ul style="list-style-type: none"> ITLB_Effectiveness MPKI
<ul style="list-style-type: none"> l2_cache_miss_ratio in L2_Cache_Effectiveness l2_cache_miss_ratio in Miss_Ratio 	$L2D_CACHE_REFILL / L2D_CACHE$	<ul style="list-style-type: none"> L2_Cache_Effectiveness Miss_Ratio
<ul style="list-style-type: none"> l2_cache_mpki in L2_Cache_Effectiveness l2_cache_mpki in MPKI 	$L2D_CACHE_REFILL / INST_RETIRED * 1000$	<ul style="list-style-type: none"> L2_Cache_Effectiveness MPKI
<ul style="list-style-type: none"> l2_tlb_miss_ratio in DTLB_Effectiveness l2_tlb_miss_ratio in ITLB_Effectiveness l2_tlb_miss_ratio in Miss_Ratio 	$L2D_TLB_REFILL / L2D_TLB$	<ul style="list-style-type: none"> DTLB_Effectiveness ITLB_Effectiveness Miss_Ratio
<ul style="list-style-type: none"> l2_tlb_mpki in DTLB_Effectiveness l2_tlb_mpki in ITLB_Effectiveness l2_tlb_mpki in MPKI 	$L2D_TLB_REFILL / INST_RETIRED * 1000$	<ul style="list-style-type: none"> DTLB_Effectiveness ITLB_Effectiveness MPKI
ll_cache_read_hit_ratio	$(LL_CACHE_RD - LL_CACHE_MISS_RD) / LL_CACHE_RD$	<ul style="list-style-type: none"> LL_Cache_Effectiveness
<ul style="list-style-type: none"> ll_cache_read_miss_ratio in LL_Cache_Effectiveness ll_cache_read_miss_ratio in Miss_Ratio 	$LL_CACHE_MISS_RD / LL_CACHE_RD$	<ul style="list-style-type: none"> LL_Cache_Effectiveness Miss_Ratio
<ul style="list-style-type: none"> ll_cache_read_mpki in LL_Cache_Effectiveness ll_cache_read_mpki in MPKI 	$LL_CACHE_MISS_RD / INST_RETIRED * 1000$	<ul style="list-style-type: none"> LL_Cache_Effectiveness MPKI
load_percentage	$LD_SPEC / INST_SPEC * 100$	<ul style="list-style-type: none"> Operation_Mix
nonsve_fp_ops_per_cycle	$FP_FIXED_OPS_SPEC / CPU_CYCLES$	<ul style="list-style-type: none"> FP_Arithmetic_Intensity
retiring	$(1 - STALL_SLOT / (CPU_CYCLES * 5)) * (OP_RETIRED / OP_SPEC) * 100$	<ul style="list-style-type: none"> Topdown_L1
scalar_fp_percentage	$VFP_SPEC / INST_SPEC * 100$	<ul style="list-style-type: none"> Operation_Mix
simd_percentage	$ASE_SPEC / INST_SPEC * 100$	<ul style="list-style-type: none"> Operation_Mix
store_percentage	$ST_SPEC / INST_SPEC * 100$	<ul style="list-style-type: none"> Operation_Mix
sve_all_percentage	$SVE_INST_SPEC / INST_SPEC * 100$	<ul style="list-style-type: none"> Operation_Mix
sve_fp_ops_per_cycle	$FP_SCALE_OPS_SPEC / CPU_CYCLES$	<ul style="list-style-type: none"> FP_Arithmetic_Intensity
sve_predicate_empty_percentage	$SVE_PRED_EMPTY_SPEC / SVE_PRED_SPEC * 100$	<ul style="list-style-type: none"> SVE_Effectiveness

Metric Name	Formula from Events	Metric Groups
sve_predicate_full_percentage	$\text{SVE_PRED_FULL_SPEC} / \text{SVE_PRED_SPEC} * 100$	<ul style="list-style-type: none"> SVE_Effectiveness
sve_predicate_partial_percentage	$\text{SVE_PRED_PARTIAL_SPEC} / \text{SVE_PRED_SPEC} * 100$	<ul style="list-style-type: none"> SVE_Effectiveness
sve_predicate_percentage	$\text{SVE_PRED_SPEC} / \text{INST_SPEC} * 100$	<ul style="list-style-type: none"> SVE_Effectiveness

4.4 PMU events lookup table for Neoverse N3

All events are listed in event code order, with the related metrics, metric groups, and functional groups. Some events are not used in the Methodology, however, they are all listed for completeness.

Summary of Events:

- Total Possible Common events: 746
- Total implemented Common events: 202
 - Common : Architectural-defined events: 158
 - Common : Implementation-defined events: 44
- Total Implemented Product ImpDef events: 12
- PMU Only Events : 12
- ETE Only Events : 2

Table 4-14: Events listed by Event Code, with related Metrics, Metric Groups, and Functional Groups

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0000, SW_INCR	-	-	<ul style="list-style-type: none"> Retired
0x0001, L1I_CACHE_REFILL	<ul style="list-style-type: none"> l1i_cache_mpki in L1I_Cache_Effectiveness l1i_cache_mpki in MPKI l1i_cache_miss_ratio in L1I_Cache_Effectiveness l1i_cache_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> L1I_Cache_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> L1I_Cache
0x0002, L1I_TLB_REFILL	<ul style="list-style-type: none"> l1i_tlb_mpki in ITLB_Effectiveness l1i_tlb_mpki in MPKI l1i_tlb_miss_ratio in ITLB_Effectiveness l1i_tlb_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> ITLB_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> TLB

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0003, L1D_CACHE_REFILL	<ul style="list-style-type: none"> l1d_cache_mpki in L1D_Cache_Effectiveness l1d_cache_mpki in MPKI l1d_cache_miss_ratio in L1D_Cache_Effectiveness l1d_cache_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> L1D_Cache_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> L1D_Cache
0x0004, L1D_CACHE	<ul style="list-style-type: none"> l1d_cache_miss_ratio in L1D_Cache_Effectiveness l1d_cache_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> L1D_Cache_Effectiveness Miss_Ratio 	<ul style="list-style-type: none"> L1D_Cache
0x0005, L1D_TLB_REFILL	<ul style="list-style-type: none"> l1d_tlb_mpki in DTLB_Effectiveness l1d_tlb_mpki in MPKI l1d_tlb_miss_ratio in DTLB_Effectiveness l1d_tlb_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> DTLB_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> TLB

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0008, INST_RETIRED	<ul style="list-style-type: none"> ipc branch_mpki in Branch_Effectiveness branch_mpki in MPKI itlb_mpki in ITLB_Effectiveness itlb_mpki in MPKI l1i_tlb_mpki in ITLB_Effectiveness l1i_tlb_mpki in MPKI dtlb_mpki in DTLB_Effectiveness dtlb_mpki in MPKI l1d_tlb_mpki in DTLB_Effectiveness l1d_tlb_mpki in MPKI l2_tlb_mpki in DTLB_Effectiveness l2_tlb_mpki in ITLB_Effectiveness l2_tlb_mpki in MPKI l1i_cache_mpki in L1I_Cache_Effectiveness l1i_cache_mpki in MPKI l1d_cache_mpki in L1D_Cache_Effectiveness l1d_cache_mpki in MPKI l2_cache_mpki in L2_Cache_Effectiveness l2_cache_mpki in MPKI ll_cache_read_mpki in LL_Cache_Effectiveness ll_cache_read_mpki in MPKI 	<ul style="list-style-type: none"> Branch_Effectiveness DTLB_Effectiveness General ITLB_Effectiveness L1D_Cache_Effectiveness L1I_Cache_Effectiveness L2_Cache_Effectiveness LL_Cache_Effectiveness MPKI 	<ul style="list-style-type: none"> Retired
0x0009, EXC_TAKEN	-	-	<ul style="list-style-type: none"> Exception
0x000A, EXC_RETURN	-	-	<ul style="list-style-type: none"> Exception
0x000B, CID_WRITE_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x000C, PC_WRITE_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x000D, BR_IMMED_RETIRED	<ul style="list-style-type: none"> branch_direct_ratio 	<ul style="list-style-type: none"> Branch_Effectiveness 	<ul style="list-style-type: none"> Retired
0x000E, BR_RETURN_RETIRED	<ul style="list-style-type: none"> branch_return_ratio 	<ul style="list-style-type: none"> Branch_Effectiveness 	<ul style="list-style-type: none"> Retired
0x0010, BR_MIS_PRED	-	-	<ul style="list-style-type: none"> Spec_Operation

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0011, CPU_CYCLES	<ul style="list-style-type: none"> frontend_stalled_cycles backend_stalled_cycles frontend_bound backend_bound retiring bad_speculation ipc 	<ul style="list-style-type: none"> Cycle_Accounting General Topdown_L1 	<ul style="list-style-type: none"> General
0x0012, BR_PRED	-	-	<ul style="list-style-type: none"> Spec_Operation
0x0013, MEM_ACCESS	-	-	<ul style="list-style-type: none"> Memory
0x0014, L1I_CACHE	<ul style="list-style-type: none"> l1i_cache_miss_ratio in L1I_Cache_Effectiveness l1i_cache_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> L1I_Cache_Effectiveness Miss_Ratio 	<ul style="list-style-type: none"> L1I_Cache
0x0015, L1D_CACHE_WB	-	-	<ul style="list-style-type: none"> L1D_Cache
0x0016, L2D_CACHE	<ul style="list-style-type: none"> l2_cache_miss_ratio in L2_Cache_Effectiveness l2_cache_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> L2_Cache_Effectiveness Miss_Ratio 	<ul style="list-style-type: none"> L2_Cache
0x0017, L2D_CACHE_REFILL	<ul style="list-style-type: none"> l2_cache_mпки in L2_Cache_Effectiveness l2_cache_mпки in MPKI l2_cache_miss_ratio in L2_Cache_Effectiveness l2_cache_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> L2_Cache_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> L2_Cache
0x0018, L2D_CACHE_WB	-	-	<ul style="list-style-type: none"> L2_Cache
0x0019, BUS_ACCESS	-	-	<ul style="list-style-type: none"> Bus
0x001B, INST_SPEC	<ul style="list-style-type: none"> load_percentage store_percentage integer_dp_percentage simd_percentage scalar_fp_percentage barrier_percentage branch_percentage crypto_percentage sve_all_percentage sve_predicate_percentage fp16_percentage fp32_percentage fp64_percentage 	<ul style="list-style-type: none"> FP_Precision_Mix Operation_Mix SVE_Effectiveness 	<ul style="list-style-type: none"> Spec_Operation
0x001C, TTBR_WRITE_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x001D, BUS_CYCLES	-	-	<ul style="list-style-type: none"> Bus

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x001E, CHAIN	-	-	<ul style="list-style-type: none"> Chain
0x0020, L2D_CACHE_ALLOCATE	-	-	<ul style="list-style-type: none"> L2_Cache
0x0021, BR_RETIRED	<ul style="list-style-type: none"> branch_direct_ratio branch_indirect_ratio branch_return_ratio branch_misprediction_ratio in Branch_Effectiveness branch_misprediction_ratio in Miss_Ratio 	<ul style="list-style-type: none"> Branch_Effectiveness Miss_Ratio 	<ul style="list-style-type: none"> Retired
0x0022, BR_MIS_PRED_RETIRED	<ul style="list-style-type: none"> branch_mпки in Branch_Effectiveness branch_mпки in MPKI branch_misprediction_ratio in Branch_Effectiveness branch_misprediction_ratio in Miss_Ratio 	<ul style="list-style-type: none"> Branch_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> Retired
0x0023, STALL_FRONTEND	<ul style="list-style-type: none"> frontend_stalled_cycles frontend_core_bound frontend_mem_bound 	<ul style="list-style-type: none"> Cycle_Accounting Topdown_Frontend 	<ul style="list-style-type: none"> Stall
0x0024, STALL_BACKEND	<ul style="list-style-type: none"> backend_stalled_cycles backend_core_bound backend_mem_bound backend_busy_bound 	<ul style="list-style-type: none"> Cycle_Accounting Topdown_Backend 	<ul style="list-style-type: none"> Stall
0x0025, L1D_TLB	<ul style="list-style-type: none"> dtlb_walk_ratio in DTLB_Effectiveness dtlb_walk_ratio in Miss_Ratio l1d_tlb_miss_ratio in DTLB_Effectiveness l1d_tlb_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> DTLB_Effectiveness Miss_Ratio 	<ul style="list-style-type: none"> TLB
0x0026, L1I_TLB	<ul style="list-style-type: none"> itlb_walk_ratio in ITLB_Effectiveness itlb_walk_ratio in Miss_Ratio l1i_tlb_miss_ratio in ITLB_Effectiveness l1i_tlb_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> ITLB_Effectiveness Miss_Ratio 	<ul style="list-style-type: none"> TLB
0x0027, L2I_CACHE	-	-	<ul style="list-style-type: none"> L2_Cache
0x0028, L2I_CACHE_REFILL	-	-	<ul style="list-style-type: none"> L2_Cache
0x0029, L3D_CACHE_ALLOCATE	-	-	<ul style="list-style-type: none"> L3_Cache
0x002A, L3D_CACHE_REFILL	-	-	<ul style="list-style-type: none"> L3_Cache
0x002B, L3D_CACHE	-	-	<ul style="list-style-type: none"> L3_Cache

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x002D, L2D_TLB_REFILL	<ul style="list-style-type: none"> I2_tlb_mpki in DTLB_Effectiveness I2_tlb_mpki in ITLB_Effectiveness I2_tlb_mpki in MPKI I2_tlb_miss_ratio in DTLB_Effectiveness I2_tlb_miss_ratio in ITLB_Effectiveness I2_tlb_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> DTLB_Effectiveness ITLB_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> TLB
0x002F, L2D_TLB	<ul style="list-style-type: none"> I2_tlb_miss_ratio in DTLB_Effectiveness I2_tlb_miss_ratio in ITLB_Effectiveness I2_tlb_miss_ratio in Miss_Ratio 	<ul style="list-style-type: none"> DTLB_Effectiveness ITLB_Effectiveness Miss_Ratio 	<ul style="list-style-type: none"> TLB
0x0031, REMOTE_ACCESS	-	-	<ul style="list-style-type: none"> Memory
0x0032, LL_CACHE	-	-	<ul style="list-style-type: none"> LL_Cache
0x0033, LL_CACHE_MISS	-	-	<ul style="list-style-type: none"> LL_Cache
0x0034, DTLB_WALK	<ul style="list-style-type: none"> dtlb_mpki in DTLB_Effectiveness dtlb_mpki in MPKI dtlb_walk_ratio in DTLB_Effectiveness dtlb_walk_ratio in Miss_Ratio 	<ul style="list-style-type: none"> DTLB_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> TLB
0x0035, ITLB_WALK	<ul style="list-style-type: none"> itlb_mpki in ITLB_Effectiveness itlb_mpki in MPKI itlb_walk_ratio in ITLB_Effectiveness itlb_walk_ratio in Miss_Ratio 	<ul style="list-style-type: none"> ITLB_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> TLB
0x0036, LL_CACHE_RD	<ul style="list-style-type: none"> ll_cache_read_miss_ratio in LL_Cache_Effectiveness ll_cache_read_miss_ratio in Miss_Ratio ll_cache_read_hit_ratio 	<ul style="list-style-type: none"> LL_Cache_Effectiveness Miss_Ratio 	<ul style="list-style-type: none"> LL_Cache
0x0037, LL_CACHE_MISS_RD	<ul style="list-style-type: none"> ll_cache_read_mpki in LL_Cache_Effectiveness ll_cache_read_mpki in MPKI ll_cache_read_miss_ratio in LL_Cache_Effectiveness ll_cache_read_miss_ratio in Miss_Ratio ll_cache_read_hit_ratio 	<ul style="list-style-type: none"> LL_Cache_Effectiveness MPKI Miss_Ratio 	<ul style="list-style-type: none"> LL_Cache
0x0039, L1D_CACHE_LMISS_RD	-	-	<ul style="list-style-type: none"> L1D_Cache

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x003A, OP_RETIRED	<ul style="list-style-type: none"> retiring bad_speculation 	<ul style="list-style-type: none"> Topdown_L1 	<ul style="list-style-type: none"> Retired
0x003B, OP_SPEC	<ul style="list-style-type: none"> retiring bad_speculation 	<ul style="list-style-type: none"> Topdown_L1 	<ul style="list-style-type: none"> Spec_Operation
0x003C, STALL	-	-	<ul style="list-style-type: none"> Stall
0x003D, STALL_SLOT_BACKEND	<ul style="list-style-type: none"> backend_bound 	<ul style="list-style-type: none"> Topdown_L1 	<ul style="list-style-type: none"> Stall
0x003E, STALL_SLOT_FRONTEND	<ul style="list-style-type: none"> frontend_bound 	<ul style="list-style-type: none"> Topdown_L1 	<ul style="list-style-type: none"> Stall
0x003F, STALL_SLOT	<ul style="list-style-type: none"> retiring bad_speculation 	<ul style="list-style-type: none"> Topdown_L1 	<ul style="list-style-type: none"> Stall
0x0040, L1D_CACHE_RD	-	-	<ul style="list-style-type: none"> L1D_Cache
0x0041, L1D_CACHE_WR	-	-	<ul style="list-style-type: none"> L1D_Cache
0x0044, L1D_CACHE_REFILL_INNER	-	-	<ul style="list-style-type: none"> L1D_Cache
0x0045, L1D_CACHE_REFILL_OUTER	-	-	<ul style="list-style-type: none"> L1D_Cache
0x0048, L1D_CACHE_INVALID	-	-	<ul style="list-style-type: none"> L1D_Cache
0x0050, L2D_CACHE_RD	-	-	<ul style="list-style-type: none"> L2_Cache
0x0051, L2D_CACHE_WR	-	-	<ul style="list-style-type: none"> L2_Cache
0x0052, L2D_CACHE_REFILL_RD	-	-	<ul style="list-style-type: none"> L2_Cache
0x0053, L2D_CACHE_REFILL_WR	-	-	<ul style="list-style-type: none"> L2_Cache
0x0056, L2D_CACHE_WB_VICTIM	-	-	<ul style="list-style-type: none"> L2_Cache
0x0057, L2D_CACHE_WB_CLEAN	-	-	<ul style="list-style-type: none"> L2_Cache
0x0058, L2D_CACHE_INVALID	-	-	<ul style="list-style-type: none"> L2_Cache
0x0060, BUS_ACCESS_RD	-	-	<ul style="list-style-type: none"> Bus
0x0061, BUS_ACCESS_WR	-	-	<ul style="list-style-type: none"> Bus
0x0066, MEM_ACCESS_RD	-	-	<ul style="list-style-type: none"> Memory
0x0067, MEM_ACCESS_WR	-	-	<ul style="list-style-type: none"> Memory
0x006E, STREX_FAIL_SPEC	-	-	<ul style="list-style-type: none"> Spec_Operation
0x006F, STREX_SPEC	-	-	<ul style="list-style-type: none"> Spec_Operation
0x0070, LD_SPEC	<ul style="list-style-type: none"> load_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation
0x0071, ST_SPEC	<ul style="list-style-type: none"> store_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation
0x0073, DP_SPEC	<ul style="list-style-type: none"> integer_dp_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation
0x0074, ASE_SPEC	<ul style="list-style-type: none"> simd_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation
0x0075, VFP_SPEC	<ul style="list-style-type: none"> scalar_fp_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation
0x0076, PC_WRITE_SPEC	<ul style="list-style-type: none"> branch_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation
0x0077, CRYPTO_SPEC	<ul style="list-style-type: none"> crypto_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation
0x007C, ISB_SPEC	<ul style="list-style-type: none"> barrier_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x007D, DSB_SPEC	<ul style="list-style-type: none"> integer_dp_percentage barrier_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation
0x007E, DMB_SPEC	<ul style="list-style-type: none"> barrier_percentage 	<ul style="list-style-type: none"> Operation_Mix 	<ul style="list-style-type: none"> Spec_Operation
0x0081, EXC_UNDEF	-	-	<ul style="list-style-type: none"> Exception
0x0082, EXC_SVC	-	-	<ul style="list-style-type: none"> Exception
0x0083, EXC_PABORT	-	-	<ul style="list-style-type: none"> Exception
0x0084, EXC_DABORT	-	-	<ul style="list-style-type: none"> Exception
0x0086, EXC_IRQ	-	-	<ul style="list-style-type: none"> Exception
0x0087, EXC_FIQ	-	-	<ul style="list-style-type: none"> Exception
0x0088, EXC_SMC	-	-	<ul style="list-style-type: none"> Exception
0x008A, EXC_HVC	-	-	<ul style="list-style-type: none"> Exception
0x008B, EXC_TRAP_PABORT	-	-	<ul style="list-style-type: none"> Exception
0x008C, EXC_TRAP_DABORT	-	-	<ul style="list-style-type: none"> Exception
0x008D, EXC_TRAP_OTHER	-	-	<ul style="list-style-type: none"> Exception
0x008E, EXC_TRAP_IRQ	-	-	<ul style="list-style-type: none"> Exception
0x008F, EXC_TRAP_FIQ	-	-	<ul style="list-style-type: none"> Exception
0x0090, RC_LD_SPEC	-	-	<ul style="list-style-type: none"> Spec_Operation
0x0091, RC_ST_SPEC	-	-	<ul style="list-style-type: none"> Spec_Operation
0x00A0, L3D_CACHE_RD	-	-	<ul style="list-style-type: none"> L3_Cache
0x015C, IMP_STALL_BACKEND_IQ_SX	-	-	-
0x015D, IMP_STALL_BACKEND_IQ_MX	-	-	-
0x015E, IMP_STALL_BACKEND_IQ_LS	-	-	-
0x015F, IMP_STALL_BACKEND_IQ_VX	-	-	-
0x0160, IMP_STALL_BACKEND_MCQ	-	-	-
0x0198, IMP_L2_CHI_CBUSY0	-	-	-
0x0199, IMP_L2_CHI_CBUSY1	-	-	-
0x019A, IMP_L2_CHI_CBUSY2	-	-	-
0x019B, IMP_L2_CHI_CBUSY3	-	-	-
0x019C, IMP_L2_CHI_CBUSY_MT	-	-	-
0x1003, IMP_STALL_BACKEND_PCRF	-	-	-
0x1005, IMP_STALL_BACKEND_RSTACK	-	-	-
0x4000, SAMPLE_POP	-	-	<ul style="list-style-type: none"> SPE
0x4001, SAMPLE_FEED	-	-	<ul style="list-style-type: none"> SPE
0x4002, SAMPLE_FILTRATE	-	-	<ul style="list-style-type: none"> SPE
0x4003, SAMPLE_COLLISION	-	-	<ul style="list-style-type: none"> SPE

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x4004, CNT_CYCLES	-	-	• General
0x4005, STALL_BACKEND_MEM	<ul style="list-style-type: none"> backend_mem_cache_bound backend_cache_l1d_bound backend_cache_l2d_bound 	• Topdown_Backend	• Stall
0x4006, L1I_CACHE_LMISS	-	-	• L1I_Cache
0x4009, L2D_CACHE_LMISS_RD	-	-	• L2_Cache
0x400A, L2I_CACHE_LMISS	-	-	• L2_Cache
0x400B, L3D_CACHE_LMISS_RD	-	-	• L3_Cache
0x400C, TRB_WRAP	-	-	• TRACE
0x400D, PMU_OVFS	-	-	• Non_PMU
0x400E, TRB_TRIG	-	-	• TRACE
0x400F, PMU_HOVFS	-	-	• Non_PMU
0x4010, TRCEXTOUT0	-	-	• TRACE
0x4011, TRCEXTOUT1	-	-	• TRACE
0x4012, TRCEXTOUT2	-	-	• TRACE
0x4013, TRCEXTOUT3	-	-	• TRACE
0x4018, CTI_TRIGOUT4	-	-	• TRACE
0x4019, CTI_TRIGOUT5	-	-	• TRACE
0x401A, CTI_TRIGOUT6	-	-	• TRACE
0x401B, CTI_TRIGOUT7	-	-	• TRACE
0x4020, LDST_ALIGN_LAT	-	-	• Memory
0x4021, LD_ALIGN_LAT	-	-	• Memory
0x4022, ST_ALIGN_LAT	-	-	• Memory
0x4024, MEM_ACCESS_CHECKED	-	-	• Memory
0x4025, MEM_ACCESS_CHECKED_RD	-	-	• Memory
0x4026, MEM_ACCESS_CHECKED_WR	-	-	• Memory
0x8005, ASE_INST_SPEC	-	-	• Spec_Operation
0x8006, SVE_INST_SPEC	• sve_all_percentage	• Operation_Mix	• SVE
0x8014, FP_HP_SPEC	• fp16_percentage	• FP_Precision_Mix	• FP_Operation
0x8018, FP_SP_SPEC	• fp32_percentage	• FP_Precision_Mix	• FP_Operation
0x801C, FP_DP_SPEC	• fp64_percentage	• FP_Precision_Mix	• FP_Operation
0x8074, SVE_PRED_SPEC	<ul style="list-style-type: none"> sve_predicate_percentage sve_predicate_full_percentage sve_predicate_partial_percentage sve_predicate_empty_percentage 	• SVE_Effectiveness	• SVE

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8075, SVE_PRED_EMPTY_SPEC	<ul style="list-style-type: none"> sve_predicate_empty_percentage 	<ul style="list-style-type: none"> SVE_Effectiveness 	<ul style="list-style-type: none"> SVE
0x8076, SVE_PRED_FULL_SPEC	<ul style="list-style-type: none"> sve_predicate_full_percentage 	<ul style="list-style-type: none"> SVE_Effectiveness 	<ul style="list-style-type: none"> SVE
0x8077, SVE_PRED_PARTIAL_SPEC	<ul style="list-style-type: none"> sve_predicate_partial_percentage 	<ul style="list-style-type: none"> SVE_Effectiveness 	<ul style="list-style-type: none"> SVE
0x8079, SVE_PRED_NOT_FULL_SPEC	-	-	<ul style="list-style-type: none"> SVE
0x80BC, SVE_LDFF_SPEC	-	-	<ul style="list-style-type: none"> SVE
0x80BD, SVE_LDFF_FAULT_SPEC	-	-	<ul style="list-style-type: none"> SVE
0x80C0, FP_SCALE_OPS_SPEC	<ul style="list-style-type: none"> sve_fp_ops_per_cycle fp_ops_per_cycle 	<ul style="list-style-type: none"> FP_Arithmetic_Intensity 	<ul style="list-style-type: none"> FP_Operation
0x80C1, FP_FIXED_OPS_SPEC	<ul style="list-style-type: none"> nonsve_fp_ops_per_cycle fp_ops_per_cycle 	<ul style="list-style-type: none"> FP_Arithmetic_Intensity 	<ul style="list-style-type: none"> FP_Operation
0x80E3, ASE_SVE_INT8_SPEC	-	-	<ul style="list-style-type: none"> SVE
0x80E7, ASE_SVE_INT16_SPEC	-	-	<ul style="list-style-type: none"> SVE
0x80EB, ASE_SVE_INT32_SPEC	-	-	<ul style="list-style-type: none"> SVE
0x80EF, ASE_SVE_INT64_SPEC	-	-	<ul style="list-style-type: none"> SVE
0x8108, BR_IMMED_TAKEN_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x810C, BR_INDNR_TAKEN_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x8110, BR_IMMED_PRED_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x8111, BR_IMMED_MIS_PRED_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x8112, BR_IND_PRED_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x8113, BR_IND_MIS_PRED_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x8114, BR_RETURN_PRED_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x8115, BR_RETURN_MIS_PRED_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x8116, BR_INDNR_PRED_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x8117, BR_INDNR_MIS_PRED_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x811C, BR_PRED_RETIRED	-	-	<ul style="list-style-type: none"> Retired
0x811D, BR_IND_RETIRED	<ul style="list-style-type: none"> branch_indirect_ratio 	<ul style="list-style-type: none"> Branch_Effectiveness 	<ul style="list-style-type: none"> Retired
0x8120, INST_FETCH_PERCYC	-	-	<ul style="list-style-type: none"> Memory
0x8121, MEM_ACCESS_RD_PERCYC	-	-	<ul style="list-style-type: none"> Memory

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8124, INST_FETCH	-	-	• Memory
0x8128, DTLB_WALK_PERCYC	-	-	• TLB
0x8129, ITLB_WALK_PERCYC	-	-	• TLB
0x812A, SAMPLE_FEED_BR	-	-	• SPE
0x812B, SAMPLE_FEED_LD	-	-	• SPE
0x812C, SAMPLE_FEED_ST	-	-	• SPE
0x812D, SAMPLE_FEED_OP	-	-	• SPE
0x812E, SAMPLE_FEED_EVENT	-	-	• SPE
0x812F, SAMPLE_FEED_LAT	-	-	• SPE
0x8134, DTLB_HWUPD	-	-	• TLB
0x8135, ITLB_HWUPD	-	-	• TLB
0x8136, DTLB_STEP	-	-	• TLB
0x8137, ITLB_STEP	-	-	• TLB
0x8138, DTLB_WALK_LARGE	-	-	• TLB
0x8139, ITLB_WALK_LARGE	-	-	• TLB
0x813A, DTLB_WALK_SMALL	-	-	• TLB
0x813B, ITLB_WALK_SMALL	-	-	• TLB
0x8140, L1D_CACHE_RW	-	-	• L1D_Cache
0x8148, L2D_CACHE_RW	-	-	• L2_Cache
0x8149, L2I_CACHE_RD	-	-	• L2_Cache
0x8152, L3D_CACHE_MISS	-	-	• L3_Cache
0x8154, L1D_CACHE_HWPRF	-	-	• L1D_Cache
0x8158, STALL_FRONTEND_MEMBOUND	<ul style="list-style-type: none"> frontend_mem_bound frontend_mem_cache_bound frontend_mem_tlb_bound 	• Topdown_Frontend	• Stall
0x8159, STALL_FRONTEND_L1I	<ul style="list-style-type: none"> frontend_mem_cache_bound frontend_cache_l1i_bound frontend_cache_l2i_bound 	• Topdown_Frontend	• Stall
0x815B, STALL_FRONTEND_MEM	<ul style="list-style-type: none"> frontend_mem_cache_bound frontend_cache_l1i_bound frontend_cache_l2i_bound 	• Topdown_Frontend	• Stall
0x815C, STALL_FRONTEND_TLB	<ul style="list-style-type: none"> frontend_mem_tlb_bound 	• Topdown_Frontend	• Stall
0x8160, STALL_FRONTEND_CPUBOUND	<ul style="list-style-type: none"> frontend_core_bound frontend_core_flush_bound frontend_core_flow_bound 	• Topdown_Frontend	• Stall
0x8161, STALL_FRONTEND_FLOW	<ul style="list-style-type: none"> frontend_core_flow_bound 	• Topdown_Frontend	• Stall
0x8162, STALL_FRONTEND_FLUSH	<ul style="list-style-type: none"> frontend_bound bad_speculation frontend_core_flush_bound 	<ul style="list-style-type: none"> Topdown_Frontend Topdown_L1 	• Stall

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8164, STALL_BACKEND_MEMBOUND	<ul style="list-style-type: none"> backend_mem_bound backend_mem_cache_bound backend_mem_tlb_bound backend_mem_store_bound 	<ul style="list-style-type: none"> Topdown_Backend 	<ul style="list-style-type: none"> Stall
0x8165, STALL_BACKEND_L1D	<ul style="list-style-type: none"> backend_mem_cache_bound backend_cache_l1d_bound backend_cache_l2d_bound 	<ul style="list-style-type: none"> Topdown_Backend 	<ul style="list-style-type: none"> Stall
0x8167, STALL_BACKEND_TLB	<ul style="list-style-type: none"> backend_mem_tlb_bound 	<ul style="list-style-type: none"> Topdown_Backend 	<ul style="list-style-type: none"> Stall
0x8168, STALL_BACKEND_ST	<ul style="list-style-type: none"> backend_mem_store_bound 	<ul style="list-style-type: none"> Topdown_Backend 	<ul style="list-style-type: none"> Stall
0x816A, STALL_BACKEND_CPUBOUND	<ul style="list-style-type: none"> backend_core_bound backend_core_rename_bound 	<ul style="list-style-type: none"> Topdown_Backend 	<ul style="list-style-type: none"> Stall
0x816B, STALL_BACKEND_BUSY	<ul style="list-style-type: none"> backend_busy_bound 	<ul style="list-style-type: none"> Topdown_Backend 	<ul style="list-style-type: none"> Stall
0x816D, STALL_BACKEND_RENAME	<ul style="list-style-type: none"> backend_core_rename_bound 	<ul style="list-style-type: none"> Topdown_Backend 	<ul style="list-style-type: none"> Stall
0x8171, CAS_NEAR_PASS	-	-	<ul style="list-style-type: none"> Spec_Operation
0x8172, CAS_NEAR_SPEC	-	-	<ul style="list-style-type: none"> Spec_Operation
0x8173, CAS_FAR_SPEC	-	-	<ul style="list-style-type: none"> Spec_Operation
0x81BC, L1D_CACHE_REFILL_HWPRF	-	-	<ul style="list-style-type: none"> L1D_Cache
0x8285, L2D_CACHE_PRF	-	-	<ul style="list-style-type: none"> L2_Cache
0x828D, L2D_CACHE_REFILL_PRF	-	-	<ul style="list-style-type: none"> L2_Cache
0x829A, LL_CACHE_REFILL	-	-	<ul style="list-style-type: none"> LL_Cache

5. Metrics by metric group in Neoverse N3

Metrics are measured using different combinations of PMU events. They are organized into groups that can be analyzed together for a use case. To calculate the metrics, two or more PMU counters are programmed with the events listed for the metric. The counters are read at the same time to determine the metric value.

Summary:

- Total metrics: 67

Metrics for Neoverse N3 are grouped into the following metric groups:

- [Topdown_L1](#), Topdown Level 1 (4 metrics)
- [Topdown_Frontend](#), Topdown Frontend (8 metrics)
- [Topdown_Backend](#), Topdown Backend (9 metrics)
- [Cycle_Accounting](#), Cycle Accounting (2 metrics)
- [General](#), General (1 metrics)
- [MPKI](#), Misses Per Kilo Instructions (10 metrics)
- [Miss_Ratio](#), Miss Ratio (10 metrics)
- [SVE_Effectiveness](#), SVE Effectiveness (4 metrics)
- [FP_Arithmetic_Intensity](#), Floating Point Arithmetic Intensity (3 metrics)
- [FP_Precision_Mix](#), Floating Point Precision (3 metrics)
- [Branch_Effectiveness](#), Branch Effectiveness (5 metrics)
- [ITLB_Effectiveness](#), Instruction TLB Effectiveness (6 metrics)
- [DTLB_Effectiveness](#), Data TLB Effectiveness (6 metrics)
- [L1I_Cache_Effectiveness](#), L1 Instruction Cache Effectiveness (2 metrics)
- [L1D_Cache_Effectiveness](#), L1 Data Cache Effectiveness (2 metrics)
- [L2_Cache_Effectiveness](#), L2 Unified Cache Effectiveness (2 metrics)
- [LL_Cache_Effectiveness](#), Last Level Cache Effectiveness (3 metrics)
- [Operation_Mix](#), Speculative Operation Mix (9 metrics)

5.1 Topdown_L1 metrics for Neoverse N3

Topdown Level 1. This metric group contains the first set of metrics to begin topdown analysis of application performance, which provide the percentage distribution of processor pipeline utilization.

Summary of metrics in Topdown_L1:

- Total metrics: 4

Table 5-1: Topdown_L1 metrics summary

Metric	Name	Description
backend_bound	Backend Bound	This metric is the percentage of total slots that were stalled due to resource constraints in the...
bad_speculation	Bad Speculation	This metric is the percentage of total slots that executed operations and didn't retire due to a...
frontend_bound	Frontend Bound	This metric is the percentage of total slots that were stalled due to resource constraints in the...
retiring	Retiring	This metric is the percentage of total slots that retired operations, which indicates cycles that...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

backend_bound, Backend Bound, metric

This metric is the percentage of total slots that were stalled due to resource constraints in the backend of the processor.

Units

This unit is expressed as percent of slots.

Formula

$$\text{STALL_SLOT_BACKEND} / (5 * \text{CPU_CYCLES}) * 100$$

Related telemetry artifacts

Events

[CPU_CYCLES](#)

[STALL_SLOT_BACKEND](#)

Metric group

[Topdown_L1](#)

Methodology

Stage 1

bad_speculation, Bad Speculation, metric

This metric is the percentage of total slots that executed operations and didn't retire due to a pipeline flush. This indicates cycles that were utilized but inefficiently.

Units

This unit is expressed as percent of slots.

Formula

$$(1 - \text{STALL_SLOT} / (5 * \text{CPU_CYCLES})) * (1 - \text{OP_RETIRED} / \text{OP_SPEC}) * 100 + \text{STALL_FRONTEND_FLUSH} / \text{CPU_CYCLES} * 100$$

Related telemetry artifacts

Events

[CPU_CYCLES](#)

[OP_RETIRED](#)

[OP_SPEC](#)

[STALL_FRONTEND_FLUSH](#)

[STALL_SLOT](#)**Metric group**[Topdown_L1](#)**Methodology**

Stage 1

frontend_bound, Frontend Bound, metric

This metric is the percentage of total slots that were stalled due to resource constraints in the frontend of the processor.

Units

This unit is expressed as percent of slots.

Formula

$$\frac{(\text{STALL_SLOT_FRONTEND} / (5 * \text{CPU_CYCLES}) - \text{STALL_FRONTEND_FLUSH} / \text{CPU_CYCLES}) * 100}{}$$
Related telemetry artifacts**Events**[CPU_CYCLES](#)[STALL_FRONTEND_FLUSH](#)[STALL_SLOT_FRONTEND](#)**Metric group**[Topdown_L1](#)**Methodology**

Stage 1

retiring, Retiring, metric

This metric is the percentage of total slots that retired operations, which indicates cycles that were utilized efficiently.

Units

This unit is expressed as percent of slots.

Formula

$$(1 - \text{STALL_SLOT} / (\text{CPU_CYCLES} * 5)) * (\text{OP_RETIRED} / \text{OP_SPEC}) * 100$$
Related telemetry artifacts**Events**[CPU_CYCLES](#)[OP_RETIRED](#)[OP_SPEC](#)[STALL_SLOT](#)**Metric group**[Topdown_L1](#)

Methodology

Stage 1

5.2 Topdown_Frontend metrics for Neoverse N3

Topdown Frontend. This metric group contains a set of metrics to analyse a frontend bound workload.

Summary of metrics in Topdown_Frontend:

- Total metrics: 8

Table 5-2: Topdown_Frontend metrics summary

Metric	Name	Description
frontend_cache_l1i_bound	Frontend Cache L1I Bound	This metric is the percentage of total cycles stalled in the frontend due to memory access...
frontend_cache_l2i_bound	Frontend Cache L2I Bound	This metric is the percentage of total cycles stalled in the frontend due to memory access...
frontend_core_bound	Frontend Core Bound	This metric is the percentage of total cycles stalled in the frontend due to frontend core...
frontend_core_flow_bound	Frontend Core Flow Bound	This metric is the percentage of total cycles stalled in the frontend as the decode unit is...
frontend_core_flush_bound	Frontend Core Flush Bound	This metric is the percentage of total cycles stalled in the frontend as the processor is...
frontend_mem_bound	Frontend Memory Bound	This metric is the percentage of total cycles stalled in the frontend due to frontend core...
frontend_mem_cache_bound	Frontend Mem Cache Bound	This metric is the percentage of total cycles stalled in the frontend due to instruction fetch...
frontend_mem_tlb_bound	Frontend Mem TLB Bound	This metric is the percentage of total cycles stalled in the frontend due to instruction fetch...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

frontend_cache_l1i_bound, Frontend Cache L1I Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to memory access latency issues caused by level 1 instruction cache misses.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_FRONTEND_L1I} / (\text{STALL_FRONTEND_L1I} + \text{STALL_FRONTEND_MEM}) * 100$$

Related telemetry artifacts**Events**

[STALL_FRONTEND_L1I](#)

STALL_FRONTEND_MEM**Metric group**

Topdown_Frontend

Methodology

Stage 1

frontend_cache_l2i_bound, Frontend Cache L2I Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to memory access latency issues caused by level 2 instruction cache misses.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_FRONTEND_MEM} / (\text{STALL_FRONTEND_L1I} + \text{STALL_FRONTEND_MEM}) * 100$$
Related telemetry artifacts**Events**

STALL_FRONTEND_L1I

STALL_FRONTEND_MEM

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_core_bound, Frontend Core Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to frontend core resource constraints not related to instruction fetch latency issues caused by memory access components.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_FRONTEND_CPUBOUND} / \text{STALL_FRONTEND} * 100$$
Related telemetry artifacts**Events**

STALL_FRONTEND

STALL_FRONTEND_CPubound

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_core_flow_bound, Frontend Core Flow Bound, metric

This metric is the percentage of total cycles stalled in the frontend as the decode unit is awaiting input from the branch prediction unit.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_FRONTEND_FLOW} / \text{STALL_FRONTEND_CPUBOUND} * 100$$

Related telemetry artifacts**Events**

[STALL_FRONTEND_CPubound](#)
[STALL_FRONTEND_FLOW](#)

Metric group

[Topdown_Frontend](#)

Methodology

Stage 1

frontend_core_flush_bound, Frontend Core Flush Bound, metric

This metric is the percentage of total cycles stalled in the frontend as the processor is recovering from a pipeline flush caused by bad speculation or other machine restesters.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_FRONTEND_FLUSH} / \text{STALL_FRONTEND_CPUBOUND} * 100$$

Related telemetry artifacts**Events**

[STALL_FRONTEND_CPubound](#)
[STALL_FRONTEND_FLUSH](#)

Metric group

[Topdown_Frontend](#)

Methodology

Stage 1

frontend_mem_bound, Frontend Memory Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to frontend core resource constraints related to the instruction fetch latency issues caused by memory access components.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_FRONTEND_MEMBOUND} / \text{STALL_FRONTEND} * 100$$

Related telemetry artifacts**Events**

STALL_FRONTEND
STALL_FRONTEND_MEMBOUND

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_mem_cache_bound, Frontend Mem Cache Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to instruction fetch latency issues caused by instruction cache misses.

Units

This unit is expressed as percent of cycles.

Formula

$$\frac{(\text{STALL_FRONTEND_L1I} + \text{STALL_FRONTEND_MEM})}{\text{STALL_FRONTEND_MEMBOUND}} * 100$$
Related telemetry artifacts**Events**

STALL_FRONTEND_L1I
STALL_FRONTEND_MEM
STALL_FRONTEND_MEMBOUND

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_mem_tlb_bound, Frontend Mem TLB Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to instruction fetch latency issues caused by instruction TLB misses.

Units

This unit is expressed as percent of cycles.

Formula

$$\frac{\text{STALL_FRONTEND_TLB}}{\text{STALL_FRONTEND_MEMBOUND}} * 100$$
Related telemetry artifacts**Events**

STALL_FRONTEND_MEMBOUND
STALL_FRONTEND_TLB

Metric group[Topdown_Frontend](#)**Methodology**

Stage 1

5.3 Topdown_Backend metrics for Neoverse N3

Topdown Backend. This metric group contains a set of metrics to analyze a backend bound workload.

Summary of metrics in Topdown_Backend:

- Total metrics: 9

Table 5-3: Topdown_Backend metrics summary

Metric	Name	Description
backend_busy_bound	Backend Busy Bound	This metric is the percentage of total cycles stalled in the backend due to issue queues being...
backend_cache_l1d_bound	Backend Cache L1D Bound	This metric is the percentage of total cycles stalled in the backend due to memory access latency...
backend_cache_l2d_bound	Backend Cache L2D Bound	This metric is the percentage of total cycles stalled in the backend due to memory access latency...
backend_core_bound	Backend Core Bound	This metric is the percentage of total cycles stalled in the backend due to backend core resource...
backend_core_rename_bound	Backend Core Rename Bound	This metric is the percentage of total cycles stalled in the backend as the rename unit registers...
backend_mem_bound	Backend Memory Bound	This metric is the percentage of total cycles stalled in the backend due to backend core resource...
backend_mem_cache_bound	Backend Memory Cache Bound	This metric is the percentage of total cycles stalled in the backend due to memory latency issues...
backend_mem_store_bound	Backend Memory Store Bound	This metric is the percentage of total cycles stalled in the frontend due to memory write pending...
backend_mem_tlb_bound	Backend Memory TLB Bound	This metric is the percentage of total cycles stalled in the backend due to memory access latency...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

backend_busy_bound, Backend Busy Bound, metric

This metric is the percentage of total cycles stalled in the backend due to issue queues being full to accept operations for execution.

Units

This unit is expressed as percent of cycles.

Formula

$\text{STALL_BACKEND_BUSY} / \text{STALL_BACKEND} * 100$

Related telemetry artifacts**Events**

[STALL_BACKEND](#)
[STALL_BACKEND_BUSY](#)

Metric group

[Topdown_Backend](#)

Methodology

Stage 1

backend_cache_l1d_bound, Backend Cache L1D Bound, metric

This metric is the percentage of total cycles stalled in the backend due to memory access latency issues caused by level 1 data cache misses.

Units

This unit is expressed as percent of cycles.

Formula

$$\frac{\text{STALL_BACKEND_L1D}}{(\text{STALL_BACKEND_L1D} + \text{STALL_BACKEND_MEM})} * 100$$

Related telemetry artifacts**Events**

[STALL_BACKEND_L1D](#)
[STALL_BACKEND_MEM](#)

Metric group

[Topdown_Backend](#)

Methodology

Stage 1

backend_cache_l2d_bound, Backend Cache L2D Bound, metric

This metric is the percentage of total cycles stalled in the backend due to memory access latency issues caused by level 2 data cache misses.

Units

This unit is expressed as percent of cycles.

Formula

$$\frac{\text{STALL_BACKEND_MEM}}{(\text{STALL_BACKEND_L1D} + \text{STALL_BACKEND_MEM})} * 100$$

Related telemetry artifacts**Events**

[STALL_BACKEND_L1D](#)
[STALL_BACKEND_MEM](#)

Metric group

[Topdown_Backend](#)

Methodology

Stage 1

backend_core_bound, Backend Core Bound, metric

This metric is the percentage of total cycles stalled in the backend due to backend core resource constraints not related to instruction fetch latency issues caused by memory access components.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_BACKEND_CPUBOUND} / \text{STALL_BACKEND} * 100$$

Related telemetry artifacts**Events**

STALL_BACKEND
STALL_BACKEND_CPubound

Metric group

Topdown_Backend

Methodology

Stage 1

backend_core_rename_bound, Backend Core Rename Bound, metric

This metric is the percentage of total cycles stalled in the backend as the rename unit registers are unavailable.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_BACKEND_RENAME} / \text{STALL_BACKEND_CPUBOUND} * 100$$

Related telemetry artifacts**Events**

STALL_BACKEND_CPubound
STALL_BACKEND_RENAME

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_bound, Backend Memory Bound, metric

This metric is the percentage of total cycles stalled in the backend due to backend core resource constraints related to memory access latency issues caused by memory access components.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_BACKEND_MEMBOUND} / \text{STALL_BACKEND} * 100$$

Related telemetry artifacts**Events**

STALL_BACKEND
STALL_BACKEND_MEMBOUND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_cache_bound, Backend Memory Cache Bound, metric

This metric is the percentage of total cycles stalled in the backend due to memory latency issues caused by data cache misses.

Units

This unit is expressed as percent of cycles.

Formula

$$(\text{STALL_BACKEND_L1D} + \text{STALL_BACKEND_MEM}) / \text{STALL_BACKEND_MEMBOUND} * 100$$

Related telemetry artifacts**Events**

STALL_BACKEND_L1D
STALL_BACKEND_MEM
STALL_BACKEND_MEMBOUND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_store_bound, Backend Memory Store Bound, metric

This metric is the percentage of total cycles stalled in the frontend due to memory write pending caused by stores stalled in the pre-commit stage.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_BACKEND_ST} / \text{STALL_BACKEND_MEMBOUND} * 100$$

Related telemetry artifacts**Events**

STALL_BACKEND_MEMBOUND

STALL_BACKEND_ST

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_tlb_bound, Backend Memory TLB Bound, metric

This metric is the percentage of total cycles stalled in the backend due to memory access latency issues caused by data TLB misses.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_BACKEND_TLB} / \text{STALL_BACKEND_MEMBOUND} * 100$$

Related telemetry artifacts

Events

STALL_BACKEND_MEMBOUND

STALL_BACKEND_TLB

Metric group

Topdown_Backend

Methodology

Stage 1

5.4 Cycle_Accounting metrics for Neoverse N3

Cycle Accounting. This metric group contains a set of metrics that measure the percentage of processor cycles stalled in either frontend or backend of the processor.

Summary of metrics in Cycle_Accounting:

- Total metrics: 2

Table 5-4: Cycle_Accounting metrics summary

Metric	Name	Description
backend_stalled_cycles	Backend Stalled Cycles	This metric is the percentage of cycles that were stalled due to resource constraints in the...
frontend_stalled_cycles	Frontend Stalled Cycles	This metric is the percentage of cycles that were stalled due to resource constraints in the...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

backend_stalled_cycles, Backend Stalled Cycles, metric

This metric is the percentage of cycles that were stalled due to resource constraints in the backend unit of the processor.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_BACKEND} / \text{CPU_CYCLES} * 100$$

Related telemetry artifacts**Events**

[CPU_CYCLES](#)
[STALL_BACKEND](#)

Metric group

[Cycle_Accounting](#)

Methodology

Stage 2

frontend_stalled_cycles, Frontend Stalled Cycles, metric

This metric is the percentage of cycles that were stalled due to resource constraints in the frontend unit of the processor.

Units

This unit is expressed as percent of cycles.

Formula

$$\text{STALL_FRONTEND} / \text{CPU_CYCLES} * 100$$

Related telemetry artifacts**Events**

[CPU_CYCLES](#)
[STALL_FRONTEND](#)

Metric group

[Cycle_Accounting](#)

Methodology

Stage 2

5.5 General metrics for Neoverse N3

General. This metric group contains general CPU metrics for performance analysis.

Summary of metrics in General:

- Total metrics: 1

Table 5-5: General metrics summary

Metric	Name	Description
ipc	Instructions Per Cycle	This metric measures the number of instructions retired per cycle.

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

ipc, Instructions Per Cycle, metric

This metric measures the number of instructions retired per cycle.

Units

This unit is expressed as per cycle.

Formula

[INST_RETIRED](#) / [CPU_CYCLES](#)

Related telemetry artifacts

Events

[CPU_CYCLES](#)

[INST_RETIRED](#)

Metric group

[General](#)

Methodology

Stage 2

5.6 MPKI metrics for Neoverse N3

Misses Per Kilo Instructions. This metric group contains metrics for different CPU resources that can be measured as misses per kilo instructions.

Summary of metrics in MPKI:

- Total metrics: 10

Table 5-6: MPKI metrics summary

Metric	Name	Description
branch_mpki	Branch MPKI	This metric measures the number of branch mispredictions per thousand instructions executed.
dtlb_mpki	DTLB MPKI	This metric measures the number of data TLB Walks per thousand instructions executed.
itlb_mpki	ITLB MPKI	This metric measures the number of instruction TLB Walks per thousand instructions executed.
l1d_cache_mpki	L1D Cache MPKI	This metric measures the number of level 1 data cache accesses missed per thousand instructions...
l1d_tlb_mpki	L1 Data TLB MPKI	This metric measures the number of level 1 data TLB accesses missed per thousand instructions...

Metric	Name	Description
l1i_cache_mpki	L1I Cache MPKI	This metric measures the number of level 1 instruction cache accesses missed per thousand...
l1i_tlb_mpki	L1 Instruction TLB MPKI	This metric measures the number of level 1 instruction TLB accesses missed per thousand...
l2_cache_mpki	L2 Cache MPKI	This metric measures the number of level 2 unified cache accesses missed per thousand...
l2_tlb_mpki	L2 Unified TLB MPKI	This metric measures the number of level 2 unified TLB accesses missed per thousand instructions...
ll_cache_read_mpki	LL Cache Read MPKI	This metric measures the number of last level cache read accesses missed per thousand...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

branch_mpki, Branch MPKI, metric

This metric measures the number of branch mispredictions per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

$\text{BR_MIS_PRED_RETIRED} / \text{INST_RETIRED} * 1000$

Related telemetry artifacts

Events

[BR_MIS_PRED_RETIRED](#)
[INST_RETIRED](#)

Metric group

[MPKI](#)
Other metric group: [Branch_Effectiveness](#)

Methodology

Stage 2

dtlb_mpki, DTLB MPKI, metric

This metric measures the number of data TLB Walks per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

$\text{DTLB_WALK} / \text{INST_RETIRED} * 1000$

Related telemetry artifacts

Events

[DTLB_WALK](#)
[INST_RETIRED](#)

Metric group

[MPKI](#)

Other metric group: [DTLB_Effectiveness](#)

Methodology

Stage 2

itlb_mpki, ITLB MPKI, metric

This metric measures the number of instruction TLB Walks per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

$\text{ITLB_WALK} / \text{INST_RETIRED} * 1000$

Related telemetry artifacts

Events

[INST_RETIRED](#)

[ITLB_WALK](#)

Metric group

[MPKI](#)

Other metric group: [ITLB_Effectiveness](#)

Methodology

Stage 2

l1d_cache_mpki, L1D Cache MPKI, metric

This metric measures the number of level 1 data cache accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

$\text{L1D_CACHE_REFILL} / \text{INST_RETIRED} * 1000$

Related telemetry artifacts

Events

[INST_RETIRED](#)

[L1D_CACHE_REFILL](#)

Metric group

[MPKI](#)

Other metric group: [L1D_Cache_Effectiveness](#)

Methodology

Stage 2

l1d_tlb_mpki, L1 Data TLB MPKI, metric

This metric measures the number of level 1 data TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula
$$\text{L1D_TLB_REFILL} / \text{INST_RETIRED} * 1000$$
Related telemetry artifacts**Events**

[INST_RETIRED](#)
[L1D_TLB_REFILL](#)

Metric group

[MPKI](#)
Other metric group: [DTLB_Effectiveness](#)

Methodology

Stage 2

L1i_cache_mпки, L1I Cache MPKI, metric

This metric measures the number of level 1 instruction cache accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula
$$\text{L1I_CACHE_REFILL} / \text{INST_RETIRED} * 1000$$
Related telemetry artifacts**Events**

[INST_RETIRED](#)
[L1I_CACHE_REFILL](#)

Metric group

[MPKI](#)
Other metric group: [L1I_Cache_Effectiveness](#)

Methodology

Stage 2

L1i_tlb_mпки, L1 Instruction TLB MPKI, metric

This metric measures the number of level 1 instruction TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula
$$\text{L1I_TLB_REFILL} / \text{INST_RETIRED} * 1000$$

Related telemetry artifacts**Events**

[INST_RETIRED](#)
[L1I_TLB_REFILL](#)

Metric group

[MPKI](#)
Other metric group: [ITLB_Effectiveness](#)

Methodology

Stage 2

I2_cache_mпки, L2 Cache MPKI, metric

This metric measures the number of level 2 unified cache accesses missed per thousand instructions executed. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a unified cache.

Units

This unit is expressed as mpki.

Formula

$$\frac{\text{L2D_CACHE_REFILL}}{\text{INST_RETIRED}} * 1000$$

Related telemetry artifacts**Events**

[INST_RETIRED](#)
[L2D_CACHE_REFILL](#)

Metric group

[MPKI](#)
Other metric group: [L2_Cache_Effectiveness](#)

Methodology

Stage 2

I2_tlb_mпки, L2 Unified TLB MPKI, metric

This metric measures the number of level 2 unified TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

$$\frac{\text{L2D_TLB_REFILL}}{\text{INST_RETIRED}} * 1000$$

Related telemetry artifacts**Events**

[INST_RETIRED](#)
[L2D_TLB_REFILL](#)

Metric group

[MPKI](#)

Other metric group: [DTLB_Effectiveness](#)

Other metric group: [ITLB_Effectiveness](#)

Methodology

Stage 2

ll_cache_read_mpki, LL Cache Read MPKI, metric

This metric measures the number of last level cache read accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

$\text{LL_CACHE_MISS_RD} / \text{INST_RETIRED} * 1000$

Related telemetry artifacts

Events

[INST_RETIRED](#)

[LL_CACHE_MISS_RD](#)

Metric group

[MPKI](#)

Other metric group: [LL_Cache_Effectiveness](#)

Methodology

Stage 2

5.7 Miss_Ratio metrics for Neoverse N3

Miss Ratio. This metric group contains metrics to measure miss ratios of different processor resources.

Summary of metrics in Miss_Ratio:

- Total metrics: 10

Table 5-7: Miss_Ratio metrics summary

Metric	Name	Description
branch_misprediction_ratio	Branch Misprediction Ratio	This metric measures the ratio of branches mispredicted to the total number of branches...
dtlb_walk_ratio	DTLB Walk Ratio	This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This...
itlb_walk_ratio	ITLB Walk Ratio	This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB...

Metric	Name	Description
l1d_cache_miss_ratio	L1D Cache Miss Ratio	This metric measures the ratio of level 1 data cache accesses missed to the total number of level...
l1d_tlb_miss_ratio	L1 Data TLB Miss Ratio	This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1...
l1i_cache_miss_ratio	L1I Cache Miss Ratio	This metric measures the ratio of level 1 instruction cache accesses missed to the total number...
l1i_tlb_miss_ratio	L1 Instruction TLB Miss Ratio	This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of...
l2_cache_miss_ratio	L2 Cache Miss Ratio	This metric measures the ratio of level 2 cache accesses missed to the total number of level 2...
l2_tlb_miss_ratio	L2 Unified TLB Miss Ratio	This metric measures the ratio of level 2 unified TLB accesses missed to the total number of...
ll_cache_read_miss_ratio	LL Cache Read Miss Ratio	This metric measures the ratio of last level cache read accesses missed to the total number of...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

branch_misprediction_ratio, Branch Misprediction Ratio, metric

This metric measures the ratio of branches mispredicted to the total number of branches architecturally executed. This gives an indication of the effectiveness of the branch prediction unit.

Units

This unit is expressed as per branch.

Formula

[BR_MIS_PRED_RETIRE](#) / [BR_RETIRE](#)

Related telemetry artifacts

Events

[BR_MIS_PRED_RETIRE](#)

[BR_RETIRE](#)

Metric group

[Miss_Ratio](#)

Other metric group: [Branch_Effectiveness](#)

Methodology

Stage 2

dtlb_walk_ratio, DTLB Walk Ratio, metric

This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This gives an indication of the effectiveness of the data TLB accesses.

Units

This unit is expressed as per tlb access.

Formula

[DTLB_WALK](#) / [L1D_TLB](#)

Related telemetry artifacts**Events**[DTLB_WALK](#)[L1D_TLB](#)**Metric group**[Miss_Ratio](#)Other metric group: [DTLB_Effectiveness](#)**Methodology**

Stage 2

itlb_walk_ratio, ITLB Walk Ratio, metric

This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB accesses. This gives an indication of the effectiveness of the instruction TLB accesses.

Units

This unit is expressed as per tlb access.

Formula
$$\text{ITLB_WALK} / \text{L1I_TLB}$$
Related telemetry artifacts**Events**[ITLB_WALK](#)[L1I_TLB](#)**Metric group**[Miss_Ratio](#)Other metric group: [ITLB_Effectiveness](#)**Methodology**

Stage 2

l1d_cache_miss_ratio, L1D Cache Miss Ratio, metric

This metric measures the ratio of level 1 data cache accesses missed to the total number of level 1 data cache accesses. This gives an indication of the effectiveness of the level 1 data cache.

Units

This unit is expressed as per cache access.

Formula
$$\text{L1D_CACHE_REFILL} / \text{L1D_CACHE}$$
Related telemetry artifacts**Events**[L1D_CACHE](#)[L1D_CACHE_REFILL](#)

Metric group[Miss_Ratio](#)Other metric group: [L1D_Cache_Effectiveness](#)**Methodology**

Stage 2

l1d_tlb_miss_ratio, L1 Data TLB Miss Ratio, metric

This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1 data TLB accesses. This gives an indication of the effectiveness of the level 1 data TLB.

Units

This unit is expressed as per tlb access.

Formula
$$\text{L1D_TLB_REFILL} / \text{L1D_TLB}$$
Related telemetry artifacts**Events**[L1D_TLB](#)[L1D_TLB_REFILL](#)**Metric group**[Miss_Ratio](#)Other metric group: [DTLB_Effectiveness](#)**Methodology**

Stage 2

l1i_cache_miss_ratio, L1I Cache Miss Ratio, metric

This metric measures the ratio of level 1 instruction cache accesses missed to the total number of level 1 instruction cache accesses. This gives an indication of the effectiveness of the level 1 instruction cache.

Units

This unit is expressed as per cache access.

Formula
$$\text{L1I_CACHE_REFILL} / \text{L1I_CACHE}$$
Related telemetry artifacts**Events**[L1I_CACHE](#)[L1I_CACHE_REFILL](#)**Metric group**[Miss_Ratio](#)Other metric group: [L1I_Cache_Effectiveness](#)

Methodology

Stage 2

l1i_tlb_miss_ratio, L1 Instruction TLB Miss Ratio, metric

This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of level 1 instruction TLB accesses. This gives an indication of the effectiveness of the level 1 instruction TLB.

Units

This unit is expressed as per tlb access.

Formula

$L1I_TLB_REFILL / L1I_TLB$

Related telemetry artifacts

Events

[L1I_TLB](#)

[L1I_TLB_REFILL](#)

Metric group

[Miss_Ratio](#)

Other metric group: [ITLB_Effectiveness](#)

Methodology

Stage 2

l2_cache_miss_ratio, L2 Cache Miss Ratio, metric

This metric measures the ratio of level 2 cache accesses missed to the total number of level 2 cache accesses. This gives an indication of the effectiveness of the level 2 cache, which is a unified cache that stores both data and instruction. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a unified cache.

Units

This unit is expressed as per cache access.

Formula

$L2D_CACHE_REFILL / L2D_CACHE$

Related telemetry artifacts

Events

[L2D_CACHE](#)

[L2D_CACHE_REFILL](#)

Metric group

[Miss_Ratio](#)

Other metric group: [L2_Cache_Effectiveness](#)

Methodology

Stage 2

L2_tlb_miss_ratio, L2 Unified TLB Miss Ratio, metric

This metric measures the ratio of level 2 unified TLB accesses missed to the total number of level 2 unified TLB accesses. This gives an indication of the effectiveness of the level 2 TLB.

Units

This unit is expressed as per tlb access.

Formula

$L2D_TLB_REFILL / L2D_TLB$

Related telemetry artifacts

Events

[L2D_TLB](#)

[L2D_TLB_REFILL](#)

Metric group

[Miss_Ratio](#)

Other metric group: [DTLB_Effectiveness](#)

Other metric group: [ITLB_Effectiveness](#)

Methodology

Stage 2

ll_cache_read_miss_ratio, LL Cache Read Miss Ratio, metric

This metric measures the ratio of last level cache read accesses missed to the total number of last level cache accesses. This gives an indication of the effectiveness of the last level cache for read traffic. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a system level cache.

Units

This unit is expressed as per cache access.

Formula

$LL_CACHE_MISS_RD / LL_CACHE_RD$

Related telemetry artifacts

Events

[LL_CACHE_MISS_RD](#)

[LL_CACHE_RD](#)

Metric group

[Miss_Ratio](#)

Other metric group: [LL_Cache_Effectiveness](#)

Methodology

Stage 2

5.8 SVE_Effectiveness metrics for Neoverse N3

SVE Effectiveness. This metric group contains metrics to evaluate the effectiveness of predicated SVE instruction execution on this processor.

Summary of metrics in SVE_Effectiveness:

- Total metrics: 4

Table 5-8: SVE_Effectiveness metrics summary

Metric	Name	Description
sve_predicate_empty_percentage	SVE Empty Predicate Percentage	This metric measures scalable vector operations with no active predicates as a percentage of sve...
sve_predicate_full_percentage	SVE Full Predicate Percentage	This metric measures scalable vector operations with all active predicates as a percentage of sve...
sve_predicate_partial_percentage	SVE Partial Predicate Percentage	This metric measures scalable vector operations with at least one active predicates as a...
sve_predicate_percentage	SVE Predicate Percentage	This metric measures scalable vector operations with predicates as a percentage of operations...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

sve_predicate_empty_percentage, SVE Empty Predicate Percentage, metric

This metric measures scalable vector operations with no active predicates as a percentage of sve predicated operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{SVE_PRED_EMPTY_SPEC} / \text{SVE_PRED_SPEC} * 100$$

Related telemetry artifacts

Events

[SVE_PRED_EMPTY_SPEC](#)

[SVE_PRED_SPEC](#)

Metric group

[SVE_Effectiveness](#)

Methodology

Stage 2

sve_predicate_full_percentage, SVE Full Predicate Percentage, metric

This metric measures scalable vector operations with all active predicates as a percentage of sve predicated operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{SVE_PRED_FULL_SPEC} / \text{SVE_PRED_SPEC} * 100$$

Related telemetry artifacts**Events**

SVE_PRED_FULL_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

sve_predicate_partial_percentage, SVE Partial Predicate Percentage, metric

This metric measures scalable vector operations with at least one active predicates as a percentage of sve predicated operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{SVE_PRED_PARTIAL_SPEC} / \text{SVE_PRED_SPEC} * 100$$

Related telemetry artifacts**Events**

SVE_PRED_PARTIAL_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

sve_predicate_percentage, SVE Predicate Percentage, metric

This metric measures scalable vector operations with predicates as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{SVE_PRED_SPEC} / \text{INST_SPEC} * 100$$

Related telemetry artifacts**Events**

INST_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

5.9 FP_Arithmetic_Intensity metrics for Neoverse N3

Floating Point Arithmetic Intensity. This metric group contains metrics to evaluate the effectiveness of floating point instruction execution on this processor.

Summary of metrics in FP_Arithmetic_Intensity:

- Total metrics: 3

Table 5-9: FP_Arithmetic_Intensity metrics summary

Metric	Name	Description
fp_ops_per_cycle	Floating Point Operations per Cycle	This metric measures floating point operations per cycle in any precision performed by any...
nonsve_fp_ops_per_cycle	Non-SVE Floating Point Operations per Cycle	This metric measures floating point operations per cycle in any precision performed by an...
sve_fp_ops_per_cycle	SVE Floating Point Operations per Cycle	This metric measures floating point operations per cycle in any precision performed by SVE...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

fp_ops_per_cycle, Floating Point Operations per Cycle, metric

This metric measures floating point operations per cycle in any precision performed by any instruction. Operations are counted by computation and by vector lanes, fused computations such as multiply-add count as twice per vector lane for example.

Units

This unit is expressed as operations per cycle.

Formula

$$(FP_SCALE_OPS_SPEC + FP_FIXED_OPS_SPEC) / CPU_CYCLES$$

Related telemetry artifacts

Events

FP_FIXED_OPS_SPEC

FP_SCALE_OPS_SPEC

Metric group

FP_Arithmetic_Intensity

Methodology

Stage 2

nonsve_fp_ops_per_cycle, Non-SVE Floating Point Operations per Cycle, metric

This metric measures floating point operations per cycle in any precision performed by an instruction that is not an SVE instruction. Operations are counted by computation and by vector lanes, fused computations such as multiply-add count as twice per vector lane for example.

Units

This unit is expressed as operations per cycle.

Formula

$\text{FP_FIXED_OPS_SPEC} / \text{CPU_CYCLES}$

Related telemetry artifacts**Events**

[FP_FIXED_OPS_SPEC](#)

Metric group

[FP_Arithmetic_Intensity](#)

Methodology

Stage 2

sve_fp_ops_per_cycle, SVE Floating Point Operations per Cycle, metric

This metric measures floating point operations per cycle in any precision performed by SVE instructions. Operations are counted by computation and by vector lanes, fused computations such as multiply-add count as twice per vector lane for example.

Units

This unit is expressed as operations per cycle.

Formula

$\text{FP_SCALE_OPS_SPEC} / \text{CPU_CYCLES}$

Related telemetry artifacts**Events**

[FP_SCALE_OPS_SPEC](#)

Metric group

[FP_Arithmetic_Intensity](#)

Methodology

Stage 2

5.10 FP_Precision_Mix metrics for Neoverse N3

Floating Point Precision. This metric group contains metrics to evaluate the precision of floating point instruction execution on this processor.

Summary of metrics in FP_Precision_Mix:

- Total metrics: 3

Table 5-10: FP_Precision_Mix metrics summary

Metric	Name	Description
fp16_percentage	Half Precision Floating Point Percentage	This metric measures half-precision floating point operations as a percentage of operations...
fp32_percentage	Single Precision Floating Point Percentage	This metric measures single-precision floating point operations as a percentage of operations...
fp64_percentage	Double Precision Floating Point Percentage	This metric measures double-precision floating point operations as a percentage of operations...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

fp16_percentage, Half Precision Floating Point Percentage, metric

This metric measures half-precision floating point operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{FP_HP_SPEC} / \text{INST_SPEC} * 100$$

Related telemetry artifacts

Events

[FP_HP_SPEC](#)
[INST_SPEC](#)

Metric group

[FP_Precision_Mix](#)

Methodology

Stage 2

fp32_percentage, Single Precision Floating Point Percentage, metric

This metric measures single-precision floating point operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{FP_SP_SPEC} / \text{INST_SPEC} * 100$$

Related telemetry artifacts

Events

[FP_SP_SPEC](#)
[INST_SPEC](#)

Metric group

[FP_Precision_Mix](#)

Methodology

Stage 2

fp64_percentage, Double Precision Floating Point Percentage, metric

This metric measures double-precision floating point operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{FP_DP_SPEC} / \text{INST_SPEC} * 100$$

Related telemetry artifacts**Events**

FP_DP_SPEC

INST_SPEC

Metric group

FP_Precision_Mix

Methodology

Stage 2

5.11 Branch_Effectiveness metrics for Neoverse N3

Branch Effectiveness. This metric group contains metrics to evaluate the effectiveness of branch instruction execution on this processor.

Summary of metrics in Branch_Effectiveness:

- Total metrics: 5

Table 5-11: Branch_Effectiveness metrics summary

Metric	Name	Description
branch_direct_ratio	Branch Direct Ratio	This metric measures the ratio of direct branches retired to the total number of branches...
branch_indirect_ratio	Branch Indirect Ratio	This metric measures the ratio of indirect branches retired, including function returns, to the...
branch_misprediction_ratio	Branch Misprediction Ratio	This metric measures the ratio of branches mispredicted to the total number of branches...
branch_mpki	Branch MPKI	This metric measures the number of branch mispredictions per thousand instructions executed.
branch_return_ratio	Branch Return Ratio	This metric measures the ratio of branches retired that are function returns to the total number...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

branch_direct_ratio, Branch Direct Ratio, metric

This metric measures the ratio of direct branches retired to the total number of branches architecturally executed.

Units

This unit is expressed as per branch.

Formula

$\text{BR_IMMED_RETIRED} / \text{BR_RETIRED}$

Related telemetry artifacts**Events**

BR_IMMED_RETIRED
 BR_RETIRED

Metric group

$\text{Branch_Effectiveness}$

Methodology

Stage 2

branch_indirect_ratio, Branch Indirect Ratio, metric

This metric measures the ratio of indirect branches retired, including function returns, to the total number of branches architecturally executed.

Units

This unit is expressed as per branch.

Formula

$\text{BR_IND_RETIRED} / \text{BR_RETIRED}$

Related telemetry artifacts**Events**

BR_IND_RETIRED
 BR_RETIRED

Metric group

$\text{Branch_Effectiveness}$

Methodology

Stage 2

branch_misprediction_ratio, Branch Misprediction Ratio, metric**

This metric measures the ratio of branches mispredicted to the total number of branches architecturally executed. This gives an indication of the effectiveness of the branch prediction unit.

Units

This unit is expressed as per branch.

Formula

$\text{BR_MIS_PRED_RETIRED} / \text{BR_RETIRED}$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[BR_MIS_PRED_RETIRE](#)

[BR_RETIRE](#)

Metric group

[Branch_Effectiveness](#)

Other metric group: [Miss_Ratio](#)

Methodology

Stage 2

branch_mpk^{}**, Branch MPKI, metric

This metric measures the number of branch mispredictions per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

$$\frac{\text{BR_MIS_PRED_RETIRE}}{\text{INST_RETIRE}} * 1000$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[BR_MIS_PRED_RETIRE](#)

[INST_RETIRE](#)

Metric group

[Branch_Effectiveness](#)

Other metric group: [MPKI](#)

Methodology

Stage 2

branch_return_ratio, Branch Return Ratio, metric

This metric measures the ratio of branches retired that are function returns to the total number of branches architecturally executed.

Units

This unit is expressed as per branch.

Formula

$$\frac{\text{BR_RETURN_RETIRE}}{\text{BR_RETIRE}}$$

Related telemetry artifacts

Events

[BR_RETIRE](#)

BR_RETURN_RETIRED

Metric group

Branch_Effectiveness

Methodology

Stage 2

5.12 ITLB_Effectiveness metrics for Neoverse N3

Instruction TLB Effectiveness. This metric group contains metrics to evaluate the effectiveness of instruction TLB on this processor.

Summary of metrics in ITLB_Effectiveness:

- Total metrics: 6

Table 5-12: ITLB_Effectiveness metrics summary

Metric	Name	Description
itlb_mпки	ITLB MPKI	This metric measures the number of instruction TLB Walks per thousand instructions executed.
itlb_walk_ratio	ITLB Walk Ratio	This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB...
l1i_tlb_miss_ratio	L1 Instruction TLB Miss Ratio	This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of...
l1i_tlb_mпки	L1 Instruction TLB MPKI	This metric measures the number of level 1 instruction TLB accesses missed per thousand...
l2_tlb_miss_ratio	L2 Unified TLB Miss Ratio	This metric measures the ratio of level 2 unified TLB accesses missed to the total number of...
l2_tlb_mпки	L2 Unified TLB MPKI	This metric measures the number of level 2 unified TLB accesses missed per thousand instructions...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

itlb_mпки**, ITLB MPKI, metric

This metric measures the number of instruction TLB Walks per thousand instructions executed.

Units

This unit is expressed as mпки.

Formula

$$ITLB_WALK / INST_RETIRED * 1000$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

INST_RETIRED

[ITLB_WALK](#)**Metric group**[ITLB_Effectiveness](#)Other metric group: [MPKI](#)**Methodology**

Stage 2

itlb_walk_ratio, ITLB Walk Ratio, metric**

This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB accesses. This gives an indication of the effectiveness of the instruction TLB accesses.

Units

This unit is expressed as per tlb access.

Formula
$$\text{ITLB_WALK} / \text{L1I_TLB}$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**[ITLB_WALK](#)[L1I_TLB](#)**Metric group**[ITLB_Effectiveness](#)Other metric group: [Miss_Ratio](#)**Methodology**

Stage 2

l1i_tlb_miss_ratio, L1 Instruction TLB Miss Ratio, metric**

This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of level 1 instruction TLB accesses. This gives an indication of the effectiveness of the level 1 instruction TLB.

Units

This unit is expressed as per tlb access.

Formula
$$\text{L1I_TLB_REFILL} / \text{L1I_TLB}$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**[L1I_TLB](#)[L1I_TLB_REFILL](#)

Metric group[ITLB_Effectiveness](#)Other metric group: [Miss_Ratio](#)**Methodology**

Stage 2

I1i_tlb_mпки, L1 Instruction TLB MPKI, metric**

This metric measures the number of level 1 instruction TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mпки.

Formula
$$\text{L1I_TLB_REFILL} / \text{INST_RETIRED} * 1000$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**[INST_RETIRED](#)[L1I_TLB_REFILL](#)**Metric group**[ITLB_Effectiveness](#)Other metric group: [MPKI](#)**Methodology**

Stage 2

I2_tlb_miss_ratio, L2 Unified TLB Miss Ratio, metric**

This metric measures the ratio of level 2 unified TLB accesses missed to the total number of level 2 unified TLB accesses. This gives an indication of the effectiveness of the level 2 TLB.

Units

This unit is expressed as per tlb access.

Formula
$$\text{L2D_TLB_REFILL} / \text{L2D_TLB}$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**[L2D_TLB](#)[L2D_TLB_REFILL](#)**Metric group**[ITLB_Effectiveness](#)

Other metric group: [DTLB_Effectiveness](#)

Other metric group: [Miss_Ratio](#)

Methodology

Stage 2

`l2_tlb_mpki`, L2 Unified TLB MPKI, metric**

This metric measures the number of level 2 unified TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

[L2D_TLB_REFILL](#) / [INST_RETIRED](#) * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[INST_RETIRED](#)

[L2D_TLB_REFILL](#)

Metric group

[ITLB_Effectiveness](#)

Other metric group: [DTLB_Effectiveness](#)

Other metric group: [MPKI](#)

Methodology

Stage 2

5.13 DTLB_Effectiveness metrics for Neoverse N3

Data TLB Effectiveness. This metric group contains metrics to evaluate the effectiveness of data TLB on this processor.

Summary of metrics in DTLB_Effectiveness:

- Total metrics: 6

Table 5-13: DTLB_Effectiveness metrics summary

Metric	Name	Description
dtlb_mpki	DTLB MPKI	This metric measures the number of data TLB Walks per thousand instructions executed.
dtlb_walk_ratio	DTLB Walk Ratio	This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This...
l1d_tlb_miss_ratio	L1 Data TLB Miss Ratio	This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1...

Metric	Name	Description
l1d_tlb_mpki	L1 Data TLB MPKI	This metric measures the number of level 1 data TLB accesses missed per thousand instructions...
l2_tlb_miss_ratio	L2 Unified TLB Miss Ratio	This metric measures the ratio of level 2 unified TLB accesses missed to the total number of...
l2_tlb_mpki	L2 Unified TLB MPKI	This metric measures the number of level 2 unified TLB accesses missed per thousand instructions...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

dtlb_mpki**, DTLB MPKI, metric

This metric measures the number of data TLB Walks per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

[DTLB_WALK](#) / [INST_RETIRED](#) * 1000

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[DTLB_WALK](#)

[INST_RETIRED](#)

Metric group

[DTLB_Effectiveness](#)

Other metric group: [MPKI](#)

Methodology

Stage 2

dtlb_walk_ratio**, DTLB Walk Ratio, metric

This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This gives an indication of the effectiveness of the data TLB accesses.

Units

This unit is expressed as per tlb access.

Formula

[DTLB_WALK](#) / [L1D_TLB](#)

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[DTLB_WALK](#)

[L1D_TLB](#)

Metric group[DTLB_Effectiveness](#)Other metric group: [Miss_Ratio](#)**Methodology**

Stage 2

l1d_tlb_miss_ratio, L1 Data TLB Miss Ratio, metric**

This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1 data TLB accesses. This gives an indication of the effectiveness of the level 1 data TLB.

Units

This unit is expressed as per tlb access.

Formula
$$\text{L1D_TLB_REFILL} / \text{L1D_TLB}$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**[L1D_TLB](#)[L1D_TLB_REFILL](#)**Metric group**[DTLB_Effectiveness](#)Other metric group: [Miss_Ratio](#)**Methodology**

Stage 2

l1d_tlb_mпки, L1 Data TLB MPKI, metric**

This metric measures the number of level 1 data TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mпки.

Formula
$$\text{L1D_TLB_REFILL} / \text{INST_RETIRED} * 1000$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**[INST_RETIRED](#)[L1D_TLB_REFILL](#)**Metric group**[DTLB_Effectiveness](#)

Other metric group: [MPKI](#)

Methodology

Stage 2

`l2_tlb_miss_ratio`*, L2 Unified TLB Miss Ratio, metric**

This metric measures the ratio of level 2 unified TLB accesses missed to the total number of level 2 unified TLB accesses. This gives an indication of the effectiveness of the level 2 TLB.

Units

This unit is expressed as per tlb access.

Formula

[L2D_TLB_REFILL](#) / [L2D_TLB](#)

*** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[L2D_TLB](#)

[L2D_TLB_REFILL](#)

Metric group

[DTLB_Effectiveness](#)

Other metric group: [ITLB_Effectiveness](#)

Other metric group: [Miss_Ratio](#)

Methodology

Stage 2

`l2_tlb_mпки`*, L2 Unified TLB MPKI, metric**

This metric measures the number of level 2 unified TLB accesses missed per thousand instructions executed.

Units

This unit is expressed as mпки.

Formula

[L2D_TLB_REFILL](#) / [INST_RETIRED](#) * 1000

*** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[INST_RETIRED](#)

[L2D_TLB_REFILL](#)

Metric group

[DTLB_Effectiveness](#)

Other metric group: [ITLB_Effectiveness](#)

Other metric group: [MPKI](#)

Methodology
Stage 2

5.14 L1I_Cache_Effectiveness metrics for Neoverse N3

L1 Instruction Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of L1 Instruction cache on this processor.

Summary of metrics in L1I_Cache_Effectiveness:

- Total metrics: 2

Table 5-14: L1I_Cache_Effectiveness metrics summary

Metric	Name	Description
l1i_cache_miss_ratio	L1I Cache Miss Ratio	This metric measures the ratio of level 1 instruction cache accesses missed to the total number...
l1i_cache_mпки	L1I Cache MPKI	This metric measures the number of level 1 instruction cache accesses missed per thousand...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

[l1i_cache_miss_ratio](#) , L1I Cache Miss Ratio, metric**

This metric measures the ratio of level 1 instruction cache accesses missed to the total number of level 1 instruction cache accesses. This gives an indication of the effectiveness of the level 1 instruction cache.

Units

This unit is expressed as per cache access.

Formula

$$\frac{\text{L1I_CACHE_REFILL}}{\text{L1I_CACHE}}$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[L1I_CACHE](#)
[L1I_CACHE_REFILL](#)

Metric group

[L1I_Cache_Effectiveness](#)
Other metric group: [Miss_Ratio](#)

Methodology
Stage 2

l1i_cache_mpki, L1I Cache MPKI, metric**

This metric measures the number of level 1 instruction cache accesses missed per thousand instructions executed.

Units

This unit is expressed as mpki.

Formula

$$\text{L1I_CACHE_REFILL} / \text{INST_RETIRED} * 1000$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[INST_RETIRED](#)
[L1I_CACHE_REFILL](#)

Metric group

[L1I_Cache_Effectiveness](#)
Other metric group: [MPKI](#)

Methodology

Stage 2

5.15 L1D_Cache_Effectiveness metrics for Neoverse N3

L1 Data Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of L1 Data Cache on this processor.

Summary of metrics in L1D_Cache_Effectiveness:

- Total metrics: 2

Table 5-15: L1D_Cache_Effectiveness metrics summary

Metric	Name	Description
l1d_cache_miss_ratio	L1D Cache Miss Ratio	This metric measures the ratio of level 1 data cache accesses missed to the total number of level...
l1d_cache_mpki	L1D Cache MPKI	This metric measures the number of level 1 data cache accesses missed per thousand instructions...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

l1d_cache_miss_ratio, L1D Cache Miss Ratio, metric**

This metric measures the ratio of level 1 data cache accesses missed to the total number of level 1 data cache accesses. This gives an indication of the effectiveness of the level 1 data cache.

Units

This unit is expressed as per cache access.

Formula

$$\text{L1D_CACHE_REFILL} / \text{L1D_CACHE}$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**

[L1D_CACHE](#)

[L1D_CACHE_REFILL](#)

Metric group

[L1D_Cache_Effectiveness](#)

Other metric group: [Miss_Ratio](#)

Methodology

Stage 2

l1d_cache_mпки, L1D Cache MPKI, metric**

This metric measures the number of level 1 data cache accesses missed per thousand instructions executed.

Units

This unit is expressed as mпки.

Formula

$$\text{L1D_CACHE_REFILL} / \text{INST_RETIRED} * 1000$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**

[INST_RETIRED](#)

[L1D_CACHE_REFILL](#)

Metric group

[L1D_Cache_Effectiveness](#)

Other metric group: [MPKI](#)

Methodology

Stage 2

5.16 L2_Cache_Effectiveness metrics for Neoverse N3

L2 Unified Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of L2 Unified Cache on this processor.

Summary of metrics in L2_Cache_Effectiveness:

- Total metrics: 2

Table 5-16: L2_Cache_Effectiveness metrics summary

Metric	Name	Description
l2_cache_miss_ratio	L2 Cache Miss Ratio	This metric measures the ratio of level 2 cache accesses missed to the total number of level 2...
l2_cache_mpki	L2 Cache MPKI	This metric measures the number of level 2 unified cache accesses missed per thousand...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

[l2_cache_miss_ratio](#)**, L2 Cache Miss Ratio, metric

This metric measures the ratio of level 2 cache accesses missed to the total number of level 2 cache accesses. This gives an indication of the effectiveness of the level 2 cache, which is a unified cache that stores both data and instruction. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a unified cache.

Units

This unit is expressed as per cache access.

Formula

[L2D_CACHE_REFILL](#) / [L2D_CACHE](#)

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts

Events

[L2D_CACHE](#)

[L2D_CACHE_REFILL](#)

Metric group

[L2_Cache_Effectiveness](#)

Other metric group: [Miss_Ratio](#)

Methodology

Stage 2

[l2_cache_mpki](#)**, L2 Cache MPKI, metric

This metric measures the number of level 2 unified cache accesses missed per thousand instructions executed. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a unified cache.

Units

This unit is expressed as mpki.

Formula

$$\text{L2D_CACHE_REFILL} / \text{INST_RETIRED} * 1000$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**

[INST_RETIRED](#)

[L2D_CACHE_REFILL](#)

Metric group

[L2_Cache_Effectiveness](#)

Other metric group: [MPKI](#)

Methodology

Stage 2

5.17 LL_Cache_Effectiveness metrics for Neoverse N3

Last Level Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of Last Level Cache on this processor.

Summary of metrics in LL_Cache_Effectiveness:

- Total metrics: 3

Table 5-17: LL_Cache_Effectiveness metrics summary

Metric	Name	Description
ll_cache_read_hit_ratio	LL Cache Read Hit Ratio	This metric measures the ratio of last level cache read accesses hit in the cache to the total...
ll_cache_read_miss_ratio	LL Cache Read Miss Ratio	This metric measures the ratio of last level cache read accesses missed to the total number of...
ll_cache_read_mпки	LL Cache Read MPKI	This metric measures the number of last level cache read accesses missed per thousand...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

ll_cache_read_hit_ratio, LL Cache Read Hit Ratio, metric

This metric measures the ratio of last level cache read accesses hit in the cache to the total number of last level cache accesses. This gives an indication of the effectiveness of the last level cache for read traffic. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a system level cache.

Units

This unit is expressed as per cache access.

Formula

$$(\text{LL_CACHE_RD} - \text{LL_CACHE_MISS_RD}) / \text{LL_CACHE_RD}$$

Related telemetry artifacts**Events**

[LL_CACHE_MISS_RD](#)

[LL_CACHE_RD](#)

Metric group

[LL_Cache_Effectiveness](#)

Methodology

Stage 2

ll_cache_read_miss_ratio, LL Cache Read Miss Ratio, metric**

This metric measures the ratio of last level cache read accesses missed to the total number of last level cache accesses. This gives an indication of the effectiveness of the last level cache for read traffic. Note that cache accesses in this cache are either data memory access or instruction fetch as this is a system level cache.

Units

This unit is expressed as per cache access.

Formula

$$\text{LL_CACHE_MISS_RD} / \text{LL_CACHE_RD}$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**

[LL_CACHE_MISS_RD](#)

[LL_CACHE_RD](#)

Metric group

[LL_Cache_Effectiveness](#)

Other metric group: [Miss_Ratio](#)

Methodology

Stage 2

ll_cache_read_mпки, LL Cache Read MPKI, metric**

This metric measures the number of last level cache read accesses missed per thousand instructions executed.

Units

This unit is expressed as mпки.

Formula

$$\text{LL_CACHE_MISS_RD} / \text{INST_RETIRED} * 1000$$

** This metric is used in multiple metric groups. See the following for more information.

Related telemetry artifacts**Events**

[INST_RETIRED](#)

[LL_CACHE_MISS_RD](#)

Metric group

[LL_Cache_Effectiveness](#)

Other metric group: [MPKI](#)

Methodology

Stage 2

5.18 Operation_Mix metrics for Neoverse N3

Speculative Operation Mix. This metric group provides the distribution of micro-operation types executed for the program.

Summary of metrics in Operation_Mix:

- Total metrics: 9

Table 5-18: Operation_Mix metrics summary

Metric	Name	Description
barrier_percentage	Barrier Operations Percentage	This metric measures instruction and data barrier operations as a percentage of operations...
branch_percentage	Branch Operations Percentage	This metric measures branch operations as a percentage of operations speculatively executed.
crypto_percentage	Crypto Operations Percentage	This metric measures crypto operations as a percentage of operations speculatively executed.
integer_dp_percentage	Integer Operations Percentage	This metric measures scalar integer operations as a percentage of operations speculatively executed.
load_percentage	Load Operations Percentage	This metric measures load operations as a percentage of operations speculatively executed.
scalar_fp_percentage	Floating Point Operations Percentage	This metric measures scalar floating point operations as a percentage of operations speculatively...
simd_percentage	Advanced SIMD Operations Percentage	This metric measures advanced SIMD operations as a percentage of total operations speculatively...
store_percentage	Store Operations Percentage	This metric measures store operations as a percentage of operations speculatively executed.
sve_all_percentage	SVE Operations (Load/Store Inclusive) Percentage	This metric measures scalable vector operations, including loads and stores, as a percentage of...

For a complete list of the metrics in Neoverse N3, see [Metrics cheat sheet for Neoverse N3](#) and [Metrics lookup table for Neoverse N3](#).

barrier_percentage, Barrier Operations Percentage, metric

This metric measures instruction and data barrier operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$(\text{ISB_SPEC} + \text{DSB_SPEC} + \text{DMB_SPEC}) / \text{INST_SPEC} * 100$$

Related telemetry artifacts

Events

[DMB_SPEC](#)

[DSB_SPEC](#)

[INST_SPEC](#)

[ISB_SPEC](#)

Metric group

[Operation_Mix](#)

Methodology

Stage 2

branch_percentage, Branch Operations Percentage, metric

This metric measures branch operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{PC_WRITE_SPEC} / \text{INST_SPEC} * 100$$

Related telemetry artifacts

Events

[INST_SPEC](#)

[PC_WRITE_SPEC](#)

Metric group

[Operation_Mix](#)

Methodology

Stage 2

crypto_percentage, Crypto Operations Percentage, metric

This metric measures crypto operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{CRYPTO_SPEC} / \text{INST_SPEC} * 100$$

Related telemetry artifacts**Events**

CRYPTO_SPEC
INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

integer_dp_percentage, Integer Operations Percentage, metric

This metric measures scalar integer operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$(\text{DP_SPEC} - \text{DSB_SPEC}) / \text{INST_SPEC} * 100$$

Related telemetry artifacts**Events**

DP_SPEC
DSB_SPEC
INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

load_percentage, Load Operations Percentage, metric

This metric measures load operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$\text{LD_SPEC} / \text{INST_SPEC} * 100$$

Related telemetry artifacts**Events**

INST_SPEC
LD_SPEC

Metric group[Operation_Mix](#)**Methodology**

Stage 2

scalar_fp_percentage, Floating Point Operations Percentage, metric

This metric measures scalar floating point operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula
$$\text{VFP_SPEC} / \text{INST_SPEC} * 100$$
Related telemetry artifacts**Events**[INST_SPEC](#)[VFP_SPEC](#)**Metric group**[Operation_Mix](#)**Methodology**

Stage 2

simd_percentage, Advanced SIMD Operations Percentage, metric

This metric measures advanced SIMD operations as a percentage of total operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula
$$\text{ASE_SPEC} / \text{INST_SPEC} * 100$$
Related telemetry artifacts**Events**[ASE_SPEC](#)[INST_SPEC](#)**Metric group**[Operation_Mix](#)**Methodology**

Stage 2

store_percentage, Store Operations Percentage, metric

This metric measures store operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$ST_SPEC / INST_SPEC * 100$$

Related telemetry artifacts**Events**

INST_SPEC

ST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

sve_all_percentage, SVE Operations (Load/Store Inclusive) Percentage, metric

This metric measures scalable vector operations, including loads and stores, as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

$$SVE_INST_SPEC / INST_SPEC * 100$$

Related telemetry artifacts**Events**

INST_SPEC

SVE_INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

6. PMU events by functional group in Neoverse N3

The Performance Monitoring Unit (PMU) collects events through an event interface from other units in the design. These events are used as triggers for event counters. Not all of the possible events are used in the Methodology, however, they are all listed for completeness.

Neoverse N3 provides the following types of PMU events:

- Total implemented Common events: 202
- Total Implemented Product ImpDef events: 12
- PMU Only events : 12
- ETE Only events : 2
- Total implemented events without groups: 12

PMU events for Neoverse N3 are grouped into the following functional groups:

- [Bus](#), BUS (4 events)
- [Chain](#), CHAIN (1 events)
- [Exception](#), EXCEPTION (15 events)
- [L1D_Cache](#), L1D CACHE (12 events)
- [L1I_Cache](#), L1I CACHE (3 events)
- [L2_Cache](#), L2 CACHE (19 events)
- [L3_Cache](#), L3 CACHE (6 events)
- [LL_Cache](#), LL CACHE (5 events)
- [Memory](#), MEMORY (13 events)
- [Retired](#), RETIRED (22 events)
- [SPE](#), SPE (10 events)
- [Spec_Operation](#), SPEC OPERATION (22 events)
- [FP_Operation](#), FP OPERATION (5 events)
- [Stall](#), STALL (21 events)
- [General](#), GENERAL (2 events)
- [TLB](#), TLB (18 events)
- [SVE](#), SVE (12 events)
- [Non_PMU](#), NON_PMU (2 events)
- [TRACE](#), TRACE (10 events)
- [Events without a functional group](#) (12 events)

6.1 Bus (BUS) events for Neoverse N3

Bus transaction related events.

Summary of events in Bus:

- Total implemented Common events: 4
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-1: Bus events summary

Code	Mnemonic	Name	Description
0x0019	BUS_ACCESS	Bus access	Counts memory transactions issued by the CPU to the external bus, including snoop requests and...
0x001D	BUS_CYCLES	Bus cycle	Counts bus cycles in the CPU. Bus cycles represent a clock cycle in which a transaction could be...
0x0060	BUS_ACCESS_RD	Bus access, read	Counts memory read transactions seen on the external bus. Each beat of data is counted individually.
0x0061	BUS_ACCESS_WR	Bus access, write	Counts memory write transactions seen on the external bus. Each beat of data is counted...

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0019 [BUS_ACCESS](#), Bus access, event

Counts memory transactions issued by the CPU to the external bus, including snoop requests and snoop responses. Each beat of data is counted individually.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Bus](#)

0x001D [BUS_CYCLES](#), Bus cycle, event

Counts bus cycles in the CPU. Bus cycles represent a clock cycle in which a transaction could be sent or received on the interface from the CPU to the external bus. Since that interface is driven at the same clock speed as the CPU, this event is a duplicate of CPU_CYCLES.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Bus](#)

0x0060 BUS_ACCESS_RD, Bus access, read, event

Counts memory read transactions seen on the external bus. Each beat of data is counted individually.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Bus](#)

0x0061 BUS_ACCESS_WR, Bus access, write, event

Counts memory write transactions seen on the external bus. Each beat of data is counted individually.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Bus](#)

6.2 Chain (CHAIN) events for Neoverse N3

Chain related events.

Summary of events in Chain:

- Total implemented Common events: 1
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-2: Chain events summary

Code	Mnemonic	Name	Description
0x001E	CHAIN	Chain a pair of event counters	For odd-numbered counters, this event increments the count by one for each overflow of the...

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x001E CHAIN, Chain a pair of event counters, event

For odd-numbered counters, this event increments the count by one for each overflow of the preceding even-numbered counter. For even-numbered counters, there is no increment. This event is used when the even/odd pairs of registers are used as a single counter.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Chain](#)

6.3 Exception (EXCEPTION) events for Neoverse N3

Exception related events.

Summary of events in Exception:

- Total implemented Common events: 15
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-3: Exception events summary

Code	Mnemonic	Name	Description
0x0009	EXC_TAKEN	Exception taken	Counts any taken architecturally visible exceptions such as IRQ, FIQ, SError, and other...
0x000A	EXC_RETURN	Instruction architecturally executed, Condition code check pass, exception return	Counts any architecturally executed exception return instructions. For example: AArch64: ERET
0x0081	EXC_UNDEF	Exception taken, other synchronous	Counts the number of synchronous exceptions which are taken locally that are due to attempting to...
0x0082	EXC_SVC	Exception taken, Supervisor Call	Counts SVC exceptions taken locally.
0x0083	EXC_PABORT	Exception taken, Instruction Abort	Counts synchronous exceptions that are taken locally and caused by Instruction Aborts.
0x0084	EXC_DABORT	Exception taken, Data Abort or SError	Counts exceptions that are taken locally and are caused by data aborts or SErrors. Conditions...
0x0086	EXC_IRQ	Exception taken, IRQ	Counts IRQ exceptions including the virtual IRQs that are taken locally.
0x0087	EXC_FIQ	Exception taken, FIQ	Counts FIQ exceptions including the virtual FIQs that are taken locally.
0x0088	EXC_SMC	Exception taken, Secure Monitor Call	Counts SMC exceptions take to EL3.
0x008A	EXC_HVC	Exception taken, Hypervisor Call	Counts HVC exceptions taken to EL2.
0x008B	EXC_TRAP_PABORT	Exception taken, Instruction Abort not Taken locally	Counts exceptions which are traps not taken locally and are caused by Instruction Aborts. For...
0x008C	EXC_TRAP_DABORT	Exception taken, Data Abort or SError not Taken locally	Counts exceptions which are traps not taken locally and are caused by Data Aborts or SError...
0x008D	EXC_TRAP_OTHER	Exception taken, other traps not Taken locally	Counts the number of synchronous trap exceptions which are not taken locally and are not SVC,...
0x008E	EXC_TRAP_IRQ	Exception taken, IRQ not Taken locally	Counts IRQ exceptions including the virtual IRQs that are not taken locally.

Code	Mnemonic	Name	Description
0x008F	EXC_TRAP_FIQ	Exception taken, FIQ not Taken locally	Counts FIQs which are not taken locally but taken from EL0, EL1, or EL2 to EL3 (which would be...

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0009 EXC_TAKEN, Exception taken, event

Counts any taken architecturally visible exceptions such as IRQ, FIQ, SError, and other synchronous exceptions. Exceptions are counted whether or not they are taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Exception](#)

0x000A EXC_RETURN, Instruction architecturally executed, Condition code check pass, exception return, event

Counts any architecturally executed exception return instructions. For example: AArch64: ERET

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Exception](#)

0x0081 EXC_UNDEF, Exception taken, other synchronous, event

Counts the number of synchronous exceptions which are taken locally that are due to attempting to execute an instruction that is **UNDEFINED**. Attempting to execute instruction bit patterns that have not been allocated. Attempting to execute instructions when they are disabled. Attempting to execute instructions at an inappropriate Exception level. Attempting to execute an instruction when the value of PSTATE.IL is 1.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Exception](#)

0x0082 EXC_SVC, Exception taken, Supervisor Call, event

Counts SVC exceptions taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Exception](#)**0x0083 EXC_PABORT, Exception taken, Instruction Abort, event**

Counts synchronous exceptions that are taken locally and caused by Instruction Aborts.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Exception](#)**0x0084 EXC_DABORT, Exception taken, Data Abort or SError, event**

Counts exceptions that are taken locally and are caused by data aborts or SErrors. Conditions that could cause those exceptions are attempting to read or write memory where the MMU generates a fault, attempting to read or write memory with a misaligned address, interrupts from the nSEI inputs and internally generated SErrors.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Exception](#)**0x0086 EXC_IRQ, Exception taken, IRQ, event**

Counts IRQ exceptions including the virtual IRQs that are taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Exception](#)**0x0087 EXC_FIQ, Exception taken, FIQ, event**

Counts FIQ exceptions including the virtual FIQs that are taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Exception](#)**0x0088 EXC_SMC, Exception taken, Secure Monitor Call, event**

Counts SMC exceptions take to EL3.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Exception

0x008A EXC_HVC, Exception taken, Hypervisor Call, event

Counts HVC exceptions taken to EL2.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Exception

0x008B EXC_TRAP_PABORT, Exception taken, Instruction Abort not Taken locally, event

Counts exceptions which are traps not taken locally and are caused by Instruction Aborts. For example, attempting to execute an instruction with a misaligned PC.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Exception

0x008C EXC_TRAP_DABORT, Exception taken, Data Abort or SError not Taken locally, event

Counts exceptions which are traps not taken locally and are caused by Data Aborts or SError interrupts. Conditions that could cause those exceptions are:

1. Attempting to read or write memory where the MMU generates a fault,
2. Attempting to read or write memory with a misaligned address,
3. Interrupts from the SEI input.
4. internally generated SErrors.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Exception

0x008D EXC_TRAP_OTHER, Exception taken, other traps not Taken locally, event

Counts the number of synchronous trap exceptions which are not taken locally and are not SVC, SMC, HVC, data aborts, Instruction Aborts, or interrupts.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Exception](#)

0x008E EXC_TRAP_IRQ, Exception taken, IRQ not Taken locally, event

Counts IRQ exceptions including the virtual IRQs that are not taken locally.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Exception](#)

0x008F EXC_TRAP_FIQ, Exception taken, FIQ not Taken locally, event

Counts FIQs which are not taken locally but taken from EL0, EL1, or EL2 to EL3 (which would be the normal behavior for FIQs when not executing in EL3).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Exception](#)

6.4 L1D_Cache (L1D CACHE) events for Neoverse N3

L1 data cache related events.

Summary of events in L1D_Cache:

- Total implemented Common events: 12
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-4: L1D_Cache events summary

Code	Mnemonic	Name	Description
0x0003	L1D_CACHE_REFILL	Level 1 data cache refill	Counts level 1 data cache refills caused by speculatively executed load or store operations that...
0x0004	L1D_CACHE	Level 1 data cache access	Counts level 1 data cache accesses from any load/store operations. Atomic operations that resolve...
0x0015	L1D_CACHE_WB	Level 1 data cache write-back	Counts write-backs of dirty data from the L1 data cache to the L2 cache. This occurs when either...

Code	Mnemonic	Name	Description
0x0039	L1D_CACHE_LMISS_RD	Level 1 data cache long-latency read miss	Counts cache line refills into the level 1 data cache from any memory read operations, that...
0x0040	L1D_CACHE_RD	Level 1 data cache access, read	Counts level 1 data cache accesses from any load operation. Atomic load operations that resolve...
0x0041	L1D_CACHE_WR	Level 1 data cache access, write	Counts level 1 data cache accesses generated by store operations. This event also counts accesses...
0x0044	L1D_CACHE_REFILL_INNER	Level 1 data cache refill, inner	Counts level 1 data cache refills where the cache line data came from caches inside the immediate...
0x0045	L1D_CACHE_REFILL_OUTER	Level 1 data cache refill, outer	Counts level 1 data cache refills for which the cache line data came from outside the immediate...
0x0048	L1D_CACHE_INVAL	Level 1 data cache invalidate	Counts each explicit invalidation of a cache line in the level 1 data cache caused by: <ul style="list-style-type: none"> Cache...
0x8140	L1D_CACHE_RW	Level 1 data cache demand access	Counts level 1 data demand cache accesses from any load or store operation. Near atomic...
0x8154	L1D_CACHE_HWPRF	Level 1 data cache hardware prefetch	Counts level 1 data cache accesses from any load/store operations generated by the hardware...
0x81BC	L1D_CACHE_REFILL_HWPRF	Level 1 data cache refill, hardware prefetch	Counts level 1 data cache refills where the cache line is requested by a hardware prefetcher.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0003 L1D_CACHE_REFILL, Level 1 data cache refill, event

Counts level 1 data cache refills caused by speculatively executed load or store operations that missed in the level 1 data cache. This event only counts one event per cache line.

Related telemetry artifacts

Metrics

[l1d_cache_mpki](#) in [L1D_Cache_Effectiveness](#)
[l1d_cache_mpki](#) in [MPKI](#)
[l1d_cache_miss_ratio](#) in [L1D_Cache_Effectiveness](#)
[l1d_cache_miss_ratio](#) in [Miss_Ratio](#)

Metric groups

[L1D_Cache_Effectiveness](#)
[MPKI](#)
[Miss_Ratio](#)

Functional groups

[L1D_Cache](#)

0x0004 L1D_CACHE, Level 1 data cache access, event

Counts level 1 data cache accesses from any load/store operations. Atomic operations that resolve in the CPUs caches (near atomic operations) counts as both a write access and read access. Each access to a cache line is counted including the multiple accesses caused by single instructions such

as LDM or STM. Each access to other level 1 data or unified memory structures, for example refill buffers, write buffers, and write-back buffers, are also counted.

Related telemetry artifacts

Metrics

[l1d_cache_miss_ratio](#) in [L1D_Cache_Effectiveness](#)

[l1d_cache_miss_ratio](#) in [Miss_Ratio](#)

Metric groups

[L1D_Cache_Effectiveness](#)

[Miss_Ratio](#)

Functional groups

[L1D_Cache](#)

0x0015 L1D_CACHE_WB, Level 1 data cache write-back, event

Counts write-backs of dirty data from the L1 data cache to the L2 cache. This occurs when either a dirty cache line is evicted from L1 data cache and allocated in the L2 cache or dirty data is written to the L2 and possibly to the next level of cache. This event counts both victim cache line evictions and cache write-backs from snoops or cache maintenance operations. The following cache operations are not counted:

1. Invalidations which do not result in data being transferred out of the L1 (such as evictions of clean data),
2. Full line writes which write to L2 without writing L1, such as write streaming mode.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

0x0039 L1D_CACHE_LMISS_RD, Level 1 data cache long-latency read miss, event

Counts cache line refills into the level 1 data cache from any memory read operations, that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

0x0040 L1D_CACHE_RD, Level 1 data cache access, read, event

Counts level 1 data cache accesses from any load operation. Atomic load operations that resolve in the CPUs caches counts as both a write access and read access.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

0x0041 L1D_CACHE_WR, Level 1 data cache access, write, event

Counts level 1 data cache accesses generated by store operations. This event also counts accesses caused by a DC ZVA (data cache zero, specified by virtual address) instruction. Near atomic operations that resolve in the CPUs caches count as a write access and read access.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

0x0044 L1D_CACHE_REFILL_INNER, Level 1 data cache refill, inner, event

Counts level 1 data cache refills where the cache line data came from caches inside the immediate cluster of the core.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

0x0045 L1D_CACHE_REFILL_OUTER, Level 1 data cache refill, outer, event

Counts level 1 data cache refills for which the cache line data came from outside the immediate cluster of the core, like an SLC in the system interconnect or DRAM.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

0x0048 L1D_CACHE_INVALID, Level 1 data cache invalidate, event

Counts each explicit invalidation of a cache line in the level 1 data cache caused by:

- Cache Maintenance Operations (CMO) that operate by a virtual address.
- Broadcast cache coherency operations from another CPU in the system.

This event does not count for the following conditions:

1. A cache refill invalidates a cache line.

2. A CMO which is executed on that CPU and invalidates a cache line specified by set/way.

Note that CMOs that operate by set/way cannot be broadcast from one CPU to another.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

0x8140 L1D_CACHE_RW, Level 1 data cache demand access, event

Counts level 1 data demand cache accesses from any load or store operation. Near atomic operations that resolve in the CPUs caches counts as both a write access and read access.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

0x8154 L1D_CACHE_HWPRF, Level 1 data cache hardware prefetch, event

Counts level 1 data cache accesses from any load/store operations generated by the hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

0x81BC L1D_CACHE_REFILL_HWPRF, Level 1 data cache refill, hardware prefetch, event

Counts level 1 data cache refills where the cache line is requested by a hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L1D_Cache](#)

6.5 L1I_Cache (L1I CACHE) events for Neoverse N3

L1 instruction cache related events.

Summary of events in L1I_Cache:

- Total implemented Common events: 3

- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-5: L1I_Cache events summary

Code	Mnemonic	Name	Description
0x0001	L1I_CACHE_REFILL	Level 1 instruction cache refill	Counts cache line refills in the level 1 instruction cache caused by a missed instruction fetch....
0x0014	L1I_CACHE	Level 1 instruction cache access	Counts instruction fetches which access the level 1 instruction cache. Instruction cache accesses...
0x4006	L1I_CACHE_LMISS	Level 1 instruction cache long-latency miss	Counts cache line refills into the level 1 instruction cache, that incurred additional latency.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0001 L1I_CACHE_REFILL, Level 1 instruction cache refill, event

Counts cache line refills in the level 1 instruction cache caused by a missed instruction fetch. Instruction fetches may include accessing multiple instructions, but the single cache line allocation is counted once.

Related telemetry artifacts

Metrics

[l1i_cache_mpki](#) in [L1I_Cache_Effectiveness](#)
[l1i_cache_mpki](#) in [MPKI](#)
[l1i_cache_miss_ratio](#) in [L1I_Cache_Effectiveness](#)
[l1i_cache_miss_ratio](#) in [Miss_Ratio](#)

Metric groups

[L1I_Cache_Effectiveness](#)
[MPKI](#)
[Miss_Ratio](#)

Functional groups

[L1I_Cache](#)

0x0014 L1I_CACHE, Level 1 instruction cache access, event

Counts instruction fetches which access the level 1 instruction cache. Instruction cache accesses caused by cache maintenance operations are not counted.

Related telemetry artifacts

Metrics

[l1i_cache_miss_ratio](#) in [L1I_Cache_Effectiveness](#)
[l1i_cache_miss_ratio](#) in [Miss_Ratio](#)

Metric groups

[L1I_Cache_Effectiveness](#)

[Miss_Ratio](#)**Functional groups**[L1I_Cache](#)**0x4006 L1I_CACHE_LMISS, Level 1 instruction cache long-latency miss, event**

Counts cache line refills into the level 1 instruction cache, that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[L1I_Cache](#)

6.6 L2_Cache (L2 CACHE) events for Neoverse N3

L2 unified cache related events.

Summary of events in L2_Cache:

- Total implemented Common events: 19
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-6: L2_Cache events summary

Code	Mnemonic	Name	Description
0x0016	L2D_CACHE	Level 2 data cache access	Counts accesses to the level 2 cache due to data accesses. Level 2 cache is a unified cache for...
0x0017	L2D_CACHE_REFILL	Level 2 data cache refill	Counts cache line refills into the level 2 cache. Level 2 cache is a unified cache for data and...
0x0018	L2D_CACHE_WB	Level 2 data cache write-back	Counts write-backs of data from the L2 cache to outside the CPU. This includes snoops to the L2...
0x0020	L2D_CACHE_ALLOCATE	Level 2 data cache allocation without refill	Counts level 2 cache line allocations that do not fetch data from outside the level 2 data or...
0x0027	L2I_CACHE	Level 2 instruction cache access	Counts accesses to the level 2 cache due to instruction accesses. Level 2 cache is a unified...
0x0028	L2I_CACHE_REFILL	Level 2 instruction cache refill	Counts cache line refills into the level 2 cache. Level 2 cache is a unified cache for data and...
0x0050	L2D_CACHE_RD	Level 2 data cache access, read	Counts level 2 data cache accesses due to memory read operations. Level 2 cache is a unified...
0x0051	L2D_CACHE_WR	Level 2 data cache access, write	Counts level 2 cache accesses due to memory write operations. Level 2 cache is a unified cache...
0x0052	L2D_CACHE_REFILL_RD	Level 2 data cache refill, read	Counts refills for memory accesses due to memory read operation counted by L2D_CACHE_RD. Level 2...

Code	Mnemonic	Name	Description
0x0053	L2D_CACHE_REFILL_WR	Level 2 data cache refill, write	Counts refills for memory accesses due to memory write operation counted by L2D_CACHE_WR. Level 2...
0x0056	L2D_CACHE_WB_VICTIM	Level 2 data cache write-back, victim	Counts evictions from the level 2 cache because of a line being allocated into the L2 cache.
0x0057	L2D_CACHE_WB_CLEAN	Level 2 data cache write-back, cleaning and coherency	Counts write-backs from the level 2 cache that are a result of either: 1. Cache maintenance...
0x0058	L2D_CACHE_INVALID	Level 2 data cache invalidate	Counts each explicit invalidation of a cache line in the level 2 cache by cache maintenance...
0x4009	L2D_CACHE_LMISS_RD	Level 2 data cache long-latency read miss	Counts cache line refills into the level 2 unified cache from any memory read operations that...
0x400A	L2I_CACHE_LMISS	Level 2 instruction cache long-latency miss	Counts cache line refills into the level 2 unified cache from any instruction read operations...
0x8148	L2D_CACHE_RW	Level 2 data cache demand access	Counts level 2 cache demand accesses from any load/store operations. Level 2 cache is a unified...
0x8149	L2I_CACHE_RD	Level 2 instruction cache demand fetch	Counts level 2 cache accesses that are due to a demand instruction cache access.
0x8285	L2D_CACHE_PRFL	Level 2 data cache, preload or prefetch hit	Counts level 2 data cache accesses from software preload or prefetch instructions or hardware...
0x828D	L2D_CACHE_REFILL_PRFL	Level 2 data cache refill, preload or prefetch hit	Counts refills due to accesses generated as a result of prefetches.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0016 L2D_CACHE, Level 2 data cache access, event

Counts accesses to the level 2 cache due to data accesses. Level 2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the first level data cache or translation resolutions due to accesses. This event also counts write back of dirty data from level 1 data cache to the L2 cache.

Related telemetry artifacts

Metrics

[l2_cache_miss_ratio](#) in [L2_Cache_Effectiveness](#)

[l2_cache_miss_ratio](#) in [Miss_Ratio](#)

Metric groups

[L2_Cache_Effectiveness](#)

[Miss_Ratio](#)

Functional groups

[L2_Cache](#)

0x0017 L2D_CACHE_REFILL, Level 2 data cache refill, event

Counts cache line refills into the level 2 cache. Level 2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

Metrics

[l2_cache_mpki](#) in [L2_Cache_Effectiveness](#)
[l2_cache_mpki](#) in [MPKI](#)
[l2_cache_miss_ratio](#) in [L2_Cache_Effectiveness](#)
[l2_cache_miss_ratio](#) in [Miss_Ratio](#)

Metric groups

[L2_Cache_Effectiveness](#)
[MPKI](#)
[Miss_Ratio](#)

Functional groups

[L2_Cache](#)

0x0018 L2D_CACHE_WB, Level 2 data cache write-back, event

Counts write-backs of data from the L2 cache to outside the CPU. This includes snoops to the L2 (from other CPUs) which return data even if the snoops cause an invalidation. L2 cache line invalidations which do not write data outside the CPU and snoops which return data from an L1 cache are not counted. Data would not be written outside the cache when invalidating a clean cache line.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x0020 L2D_CACHE_ALLOCATE, Level 2 data cache allocation without refill, event

Counts level 2 cache line allocates that do not fetch data from outside the level 2 data or unified cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x0027 L2I_CACHE, Level 2 instruction cache access, event

Counts accesses to the level 2 cache due to instruction accesses. Level 2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the first level instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[L2_Cache](#)**0x0028 L2I_CACHE_REFILL, Level 2 instruction cache refill, event**

Counts cache line refills into the level 2 cache. Level 2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the level 1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[L2_Cache](#)**0x0050 L2D_CACHE_RD, Level 2 data cache access, read, event**

Counts level 2 data cache accesses due to memory read operations. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[L2_Cache](#)**0x0051 L2D_CACHE_WR, Level 2 data cache access, write, event**

Counts level 2 cache accesses due to memory write operations. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[L2_Cache](#)**0x0052 L2D_CACHE_REFILL_RD, Level 2 data cache refill, read, event**

Counts refills for memory accesses due to memory read operation counted by L2D_CACHE_RD. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[L2_Cache](#)

0x0053 L2D_CACHE_REFILL_WR, Level 2 data cache refill, write, event

Counts refills for memory accesses due to memory write operation counted by L2D_CACHE_WR. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x0056 L2D_CACHE_WB_VICTIM, Level 2 data cache write-back, victim, event

Counts evictions from the level 2 cache because of a line being allocated into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x0057 L2D_CACHE_WB_CLEAN, Level 2 data cache write-back, cleaning and coherency, event

Counts write-backs from the level 2 cache that are a result of either:

1. Cache maintenance operations,
2. Snoop responses or,
3. Direct cache transfers to another CPU due to a forwarding snoop request.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x0058 L2D_CACHE_INVALID, Level 2 data cache invalidate, event

Counts each explicit invalidation of a cache line in the level 2 cache by cache maintenance operations that operate by a virtual address, or by external coherency operations. This event does not count if either:

1. A cache refill invalidates a cache line or,
2. A Cache Maintenance Operation (CMO), which invalidates a cache line specified by set/way, is executed on that CPU.

CMOs that operate by set/way cannot be broadcast from one CPU to another.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x4009 L2D_CACHE_LMISS_RD, Level 2 data cache long-latency read miss, event

Counts cache line refills into the level 2 unified cache from any memory read operations that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x400A L2I_CACHE_LMISS, Level 2 instruction cache long-latency miss, event

Counts cache line refills into the level 2 unified cache from any instruction read operations that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x8148 L2D_CACHE_RW, Level 2 data cache demand access, event

Counts level 2 cache demand accesses from any load/store operations. Level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the level 1 data cache or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x8149 L2I_CACHE_RD, Level 2 instruction cache demand fetch, event

Counts level 2 cache accesses that are due to a demand instruction cache access.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x8285 L2D_CACHE_PRF, Level 2 data cache, preload or prefetch hit, event

Counts level 2 data cache accesses from software preload or prefetch instructions or hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

0x828D L2D_CACHE_REFILL_PRF, Level 2 data cache refill, preload or prefetch hit, event

Counts refills due to accesses generated as a result of prefetches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L2_Cache](#)

6.7 L3_Cache (L3 CACHE) events for Neoverse N3

L3 unified cache related events.

Summary of events in L3_Cache:

- Total implemented Common events: 6
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-7: L3_Cache events summary

Code	Mnemonic	Name	Description
0x0029	L3D_CACHE_ALLOCATE	Level 3 data cache allocation without refill	Counts level 3 cache line allocates that do not fetch data from outside the level 3 data or...
0x002A	L3D_CACHE_REFILL	Level 3 data cache refill	Counts level 3 accesses that receive data from outside the L3 cache.
0x002B	L3D_CACHE	Level 3 data cache access	Counts level 3 cache accesses. Level 3 cache is a unified cache for data and instruction...
0x00A0	L3D_CACHE_RD	Level 3 data cache access, read	Counts level 3 cache accesses caused by any memory read operation. Level 3 cache is a unified...
0x400B	L3D_CACHE_LMISS_RD	Level 3 data cache long-latency read miss	Counts any cache line refill into the level 3 cache from memory read operations that incurred...
0x8152	L3D_CACHE_MISS	Level 3 data cache demand access miss	Counts level 3 cache accesses that missed in the level 3 cache.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0029 L3D_CACHE_ALLOCATE, Level 3 data cache allocation without refill, event

Counts level 3 cache line allocates that do not fetch data from outside the level 3 data or unified cache. For example, allocates due to streaming stores.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L3_Cache](#)

0x002A L3D_CACHE_REFILL, Level 3 data cache refill, event

Counts level 3 accesses that receive data from outside the L3 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L3_Cache](#)

0x002B L3D_CACHE, Level 3 data cache access, event

Counts level 3 cache accesses. Level 3 cache is a unified cache for data and instruction accesses. Accesses are for misses in the lower level caches or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L3_Cache](#)

0x00A0 L3D_CACHE_RD, Level 3 data cache access, read, event

Counts level 3 cache accesses caused by any memory read operation. Level 3 cache is a unified cache for data and instruction accesses. Accesses are for misses in the lower level caches or translation resolutions due to accesses.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L3_Cache](#)

0x400B L3D_CACHE_LMISS_RD, Level 3 data cache long-latency read miss, event

Counts any cache line refill into the level 3 cache from memory read operations that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L3_Cache](#)

0x8152 L3D_CACHE_MISS, Level 3 data cache demand access miss, event

Counts level 3 cache accesses that missed in the level 3 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[L3_Cache](#)

6.8 LL_Cache (LL CACHE) events for Neoverse N3

Last Level Cache related events.

Summary of events in LL_Cache:

- Total implemented Common events: 5
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-8: LL_Cache events summary

Code	Mnemonic	Name	Description
0x0032	LL_CACHE	Last level cache access	Counts transactions that were returned from outside the core cluster. This event counts...
0x0033	LL_CACHE_MISS	Last level cache miss	Counts transactions that were returned from outside the core cluster and missed in the last level...
0x0036	LL_CACHE_RD	Last level cache access, read	Counts read transactions that were returned from outside the core cluster. This event counts for...
0x0037	LL_CACHE_MISS_RD	Last level cache miss, read	Counts read transactions that were returned from outside the core cluster but missed in the...
0x829A	LL_CACHE_REFILL	Last level cache refill	Counts last level accesses that receive data from outside the last level cache.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0032 LL_CACHE, Last level cache access, event

Counts transactions that were returned from outside the core cluster. This event counts transactions for external last level cache when the system register CPUECTLR.EXTLLC bit is set, otherwise it counts transactions for L3 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[LL_Cache](#)

0x0033 LL_CACHE_MISS, Last level cache miss, event

Counts transactions that were returned from outside the core cluster and missed in the last level cache

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[LL_Cache](#)

0x0036 LL_CACHE_RD, Last level cache access, read, event

Counts read transactions that were returned from outside the core cluster. This event counts for external last level cache when the system register CPUECTLR.EXTLLC bit is set, otherwise it counts for the L3 cache. This event counts read transactions returned from outside the core if those transactions are either hit in the system level cache or missed in the SLC and are returned from any other external sources.

Related telemetry artifacts**Metrics**

[ll_cache_read_miss_ratio](#) in [LL_Cache_Effectiveness](#)

[ll_cache_read_miss_ratio](#) in [Miss_Ratio](#)

[ll_cache_read_hit_ratio](#)

Metric groups

[LL_Cache_Effectiveness](#)

[Miss_Ratio](#)

Functional groups

[LL_Cache](#)

0x0037 LL_CACHE_MISS_RD, Last level cache miss, read, event

Counts read transactions that were returned from outside the core cluster but missed in the system level cache. This event counts for external last level cache when the system register CPUECTLR.EXTLLC bit is set, otherwise it counts for L3 cache. This event counts read transactions returned from outside the core if those transactions are missed in the System level Cache. The data

source of the transaction is indicated by a field in the CHI transaction returning to the CPU. This event does not count reads caused by cache maintenance operations.

Related telemetry artifacts

Metrics

ll_cache_read_mpki in LL_Cache_Effectiveness
ll_cache_read_mpki in MPKI
ll_cache_read_miss_ratio in LL_Cache_Effectiveness
ll_cache_read_miss_ratio in Miss_Ratio
ll_cache_read_hit_ratio

Metric groups

LL_Cache_Effectiveness
MPKI
Miss_Ratio

Functional groups

LL_Cache

0x829A LL_CACHE_REFILL, Last level cache refill, event

Counts last level accesses that receive data from outside the last level cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

LL_Cache

6.9 Memory (MEMORY) events for Neoverse N3

Memory system related events.

Summary of events in Memory:

- Total implemented Common events: 13
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-9: Memory events summary

Code	Mnemonic	Name	Description
0x0013	MEM_ACCESS	Data memory access	Counts memory accesses issued by the CPU load store unit, where those accesses are issued due to...
0x0031	REMOTE_ACCESS	Access to another socket in a multi-socket system	Counts accesses to another chip, which is implemented as a different CMN mesh in the system. If...

Code	Mnemonic	Name	Description
0x0066	MEM_ACCESS_RD	Data memory access, read	Counts memory accesses issued by the CPU due to load operations. The event counts any memory load...
0x0067	MEM_ACCESS_WR	Data memory access, write	Counts memory accesses issued by the CPU due to store operations. The event counts any memory...
0x4020	LDST_ALIGN_LAT	Access with additional latency from alignment	Counts the number of memory read and write accesses in a cycle that incurred additional latency...
0x4021	LD_ALIGN_LAT	Load with additional latency from alignment	Counts the number of memory read accesses in a cycle that incurred additional latency, due to the...
0x4022	ST_ALIGN_LAT	Store with additional latency from alignment	Counts the number of memory write access in a cycle that incurred additional latency, due to the...
0x4024	MEM_ACCESS_CHECKED	Checked data memory access	Counts the number of memory read and write accesses counted by MEM_ACCESS that are tag checked by...
0x4025	MEM_ACCESS_CHECKED_RD	Checked data memory access, read	Counts the number of memory read accesses in a cycle that are tag checked by the Memory Tagging...
0x4026	MEM_ACCESS_CHECKED_WR	Checked data memory access, write	Counts the number of memory write accesses in a cycle that is tag checked by the Memory Tagging...
0x8120	INST_FETCH_PERCYC	Event in progress, INST FETCH	Counts number of instruction fetches outstanding per cycle, which will provide an average latency...
0x8121	MEM_ACCESS_RD_PERCYC	Event in progress, MEM ACCESS RD	Counts the number of outstanding loads or memory read accesses per cycle.
0x8124	INST_FETCH	Instruction memory access	Counts Instruction memory accesses that the PE makes.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0013 MEM_ACCESS, Data memory access, event

Counts memory accesses issued by the CPU load store unit, where those accesses are issued due to load or store operations. This event counts memory accesses no matter whether the data is received from any level of cache hierarchy or external memory. If memory accesses are broken up into smaller transactions than what were specified in the load or store instructions, then the event counts those smaller memory transactions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Memory](#)

0x0031 REMOTE_ACCESS, Access to another socket in a multi-socket system, event

Counts accesses to another chip, which is implemented as a different CMN mesh in the system. If the CHI bus response back to the core indicates that the data source is from another chip (mesh), then the counter is updated. If no data is returned, even if the system snoops another chip/mesh, then the counter is not updated.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Memory

0x0066 MEM_ACCESS_RD, Data memory access, read, event

Counts memory accesses issued by the CPU due to load operations. The event counts any memory load access, no matter whether the data is received from any level of cache hierarchy or external memory. The event also counts atomic load operations. If memory accesses are broken up by the load/store unit into smaller transactions that are issued by the bus interface, then the event counts those smaller transactions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Memory

0x0067 MEM_ACCESS_WR, Data memory access, write, event

Counts memory accesses issued by the CPU due to store operations. The event counts any memory store access, no matter whether the data is located in any level of cache or external memory. The event also counts atomic load and store operations. If memory accesses are broken up by the load/store unit into smaller transactions that are issued by the bus interface, then the event counts those smaller transactions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Memory

0x4020 LDST_ALIGN_LAT, Access with additional latency from alignment, event

Counts the number of memory read and write accesses in a cycle that incurred additional latency, due to the alignment of the address and the size of data being accessed, which results in store crossing a single cache line.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Memory

0x4021 LD_ALIGN_LAT, Load with additional latency from alignment, event

Counts the number of memory read accesses in a cycle that incurred additional latency, due to the alignment of the address and size of data being accessed, which results in load crossing a single cache line.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Memory](#)

0x4022 ST_ALIGN_LAT, Store with additional latency from alignment, event

Counts the number of memory write access in a cycle that incurred additional latency, due to the alignment of the address and size of data being accessed incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Memory](#)

0x4024 MEM_ACCESS_CHECKED, Checked data memory access, event

Counts the number of memory read and write accesses counted by MEM_ACCESS that are tag checked by the Memory Tagging Extension (MTE). This event is implemented as the sum of MEM_ACCESS_CHECKED_RD and MEM_ACCESS_CHECKED_WR

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Memory](#)

0x4025 MEM_ACCESS_CHECKED_RD, Checked data memory access, read, event

Counts the number of memory read accesses in a cycle that are tag checked by the Memory Tagging Extension (MTE).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Memory](#)

0x4026 MEM_ACCESS_CHECKED_WR, Checked data memory access, write, event

Counts the number of memory write accesses in a cycle that is tag checked by the Memory Tagging Extension (MTE).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Memory](#)

0x8120 INST_FETCH_PERCYC, Event in progress, INST FETCH, event

Counts number of instruction fetches outstanding per cycle, which will provide an average latency of instruction fetch.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Memory

0x8121 MEM_ACCESS_RD_PERCYC, Event in progress, MEM ACCESS RD, event

Counts the number of outstanding loads or memory read accesses per cycle.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Memory

0x8124 INST_FETCH, Instruction memory access, event

Counts Instruction memory accesses that the PE makes.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Memory

6.10 Retired (RETIRED) events for Neoverse N3

Retired instruction and operation events.

Summary of events in Retired:

- Total implemented Common events: 22
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-10: Retired events summary

Code	Mnemonic	Name	Description
0x0000	SW_INCR	Instruction architecturally executed, Condition code check pass, software increment	Counts software writes to the PMSWINC_ELO (software PMU increment) register. The PMSWINC_ELO...

Code	Mnemonic	Name	Description
0x0008	INST_RETIRED	Instruction architecturally executed	Counts instructions that have been architecturally executed.
0x000B	CID_WRITE_RETIRED	Instruction architecturally executed, Condition code check pass, write to CONTEXTIDR	Counts architecturally executed writes to the CONTEXTIDR_EL1 register, which usually contain the...
0x000C	PC_WRITE_RETIRED	Instruction architecturally executed, Condition code check pass, Software change of the PC	Counts branch instructions that caused a change of Program Counter, which effectively causes a...
0x000D	BR_IMMED_RETIRED	Branch instruction architecturally executed, immediate	Counts architecturally executed direct branches.
0x000E	BR_RETURN_RETIRED	Branch instruction architecturally executed, procedure return, taken	Counts architecturally executed procedure returns.
0x001C	TTBR_WRITE_RETIRED	Instruction architecturally executed, Condition code check pass, write to TTBR	Counts architectural writes to TTBR0/1_EL1. If virtualization host extensions are enabled (by...
0x0021	BR_RETIRED	Instruction architecturally executed, branch	Counts architecturally executed branches, whether the branch is taken or not. Instructions that...
0x0022	BR_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted	Counts branches counted by BR_RETIRED which were mispredicted and caused a pipeline flush.
0x003A	OP_RETIRED	Micro-operation architecturally executed	Counts micro-operations that are architecturally executed. This is a count of number of...
0x8108	BR_IMMED_TAKEN_RETIRED	Branch instruction architecturally executed, immediate, taken	Counts architecturally executed direct branches that were taken.
0x810C	BR_INDNR_TAKEN_RETIRED	Branch instruction architecturally executed, indirect excluding procedure return, taken	Counts architecturally executed indirect branches excluding procedure returns that were taken.
0x8110	BR_IMMED_PRED_RETIRED	Branch instruction architecturally executed, predicted immediate	Counts architecturally executed direct branches that were correctly predicted.
0x8111	BR_IMMED_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted immediate	Counts architecturally executed direct branches that were mispredicted and caused a pipeline flush.
0x8112	BR_IND_PRED_RETIRED	Branch instruction architecturally executed, predicted indirect	Counts architecturally executed indirect branches including procedure returns that were correctly...
0x8113	BR_IND_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted indirect	Counts architecturally executed indirect branches including procedure returns that were...
0x8114	BR_RETURN_PRED_RETIRED	Branch instruction architecturally executed, predicted procedure return	Counts architecturally executed procedure returns that were correctly predicted.
0x8115	BR_RETURN_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted procedure return	Counts architecturally executed procedure returns that were mispredicted and caused a pipeline...
0x8116	BR_INDNR_PRED_RETIRED	Branch instruction architecturally executed, predicted indirect excluding procedure return	Counts architecturally executed indirect branches excluding procedure returns that were correctly...
0x8117	BR_INDNR_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted indirect excluding procedure return	Counts architecturally executed indirect branches excluding procedure returns that were...

Code	Mnemonic	Name	Description
0x811C	BR_PRED_RETIRED	Branch instruction architecturally executed, predicted branch	Counts branch instructions counted by BR_RETIRED which were correctly predicted.
0x811D	BR_IND_RETIRED	Instruction architecturally executed, indirect branch	Counts architecturally executed indirect branches including procedure returns.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0000 SW_INCR, Instruction architecturally executed, Condition code check pass, software increment, event

Counts software writes to the PMSWINC_ELO (software PMU increment) register. The PMSWINC_ELO register is a manually updated counter for use by application software.

This event could be used to measure any user program event, such as accesses to a particular data structure (by writing to the PMSWINC_ELO register each time the data structure is accessed).

To use the PMSWINC_ELO register and event, developers must insert instructions that write to the PMSWINC_ELO register into the source code.

Since the SW_INCR event records writes to the PMSWINC_ELO register, there is no need to do a read/increment/write sequence to the PMSWINC_ELO register.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x0008 INST_RETIRED, Instruction architecturally executed, event

Counts instructions that have been architecturally executed.

Related telemetry artifacts

Metrics

[ipc](#)
[branch_mpki](#) in [Branch_Effectiveness](#)
[branch_mpki](#) in [MPKI](#)
[itlb_mpki](#) in [ITLB_Effectiveness](#)
[itlb_mpki](#) in [MPKI](#)
[l1i_tlb_mpki](#) in [ITLB_Effectiveness](#)
[l1i_tlb_mpki](#) in [MPKI](#)
[dtlb_mpki](#) in [DTLB_Effectiveness](#)
[dtlb_mpki](#) in [MPKI](#)
[l1d_tlb_mpki](#) in [DTLB_Effectiveness](#)
[l1d_tlb_mpki](#) in [MPKI](#)
[l2_tlb_mpki](#) in [DTLB_Effectiveness](#)

[l2_tlb_mpki in ITLB_Effectiveness](#)
[l2_tlb_mpki in MPKI](#)
[l1i_cache_mpki in L1I_Cache_Effectiveness](#)
[l1i_cache_mpki in MPKI](#)
[l1d_cache_mpki in L1D_Cache_Effectiveness](#)
[l1d_cache_mpki in MPKI](#)
[l2_cache_mpki in L2_Cache_Effectiveness](#)
[l2_cache_mpki in MPKI](#)
[ll_cache_read_mpki in LL_Cache_Effectiveness](#)
[ll_cache_read_mpki in MPKI](#)

Metric groups

[Branch_Effectiveness](#)
[DTLB_Effectiveness](#)
[General](#)
[ITLB_Effectiveness](#)
[L1D_Cache_Effectiveness](#)
[L1I_Cache_Effectiveness](#)
[L2_Cache_Effectiveness](#)
[LL_Cache_Effectiveness](#)
[MPKI](#)

Functional groups

[Retired](#)

0x000b CID_WRITE_RETIRED, Instruction architecturally executed, Condition code check pass, write to CONTEXTIDR, event

Counts architecturally executed writes to the CONTEXTIDR_EL1 register, which usually contain the kernel PID and can be output with hardware trace.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x000c PC_WRITE_RETIRED, Instruction architecturally executed, Condition code check pass, Software change of the PC, event

Counts branch instructions that caused a change of Program Counter, which effectively causes a change in the control flow of the program.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Retired](#)**0x000D BR_IMMED_RETIRED, Branch instruction architecturally executed, immediate, event**

Counts architecturally executed direct branches.

Related telemetry artifacts**Metrics**[branch_direct_ratio](#)**Metric groups**[Branch_Effectiveness](#)**Functional groups**[Retired](#)**0x000E BR_RETURN_RETIRED, Branch instruction architecturally executed, procedure return, taken, event**

Counts architecturally executed procedure returns.

Related telemetry artifacts**Metrics**[branch_return_ratio](#)**Metric groups**[Branch_Effectiveness](#)**Functional groups**[Retired](#)**0x001C TTBR_WRITE_RETIRED, Instruction architecturally executed, Condition code check pass, write to TTBR, event**

Counts architectural writes to TTBR0/1_EL1. If virtualization host extensions are enabled (by setting the HCR_EL2.E2H bit to 1), then accesses to TTBR0/1_EL1 that are redirected to TTBR0/1_EL2, or accesses to TTBR0/1_EL12, are counted. TTBRn registers are typically updated when the kernel is swapping user-space threads or applications.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Retired](#)**0x0021 BR_RETIRED, Instruction architecturally executed, branch, event**

Counts architecturally executed branches, whether the branch is taken or not. Instructions that explicitly write to the PC are also counted. Note that exception generating instructions, exception return instructions and context synchronization instructions are not counted.

Related telemetry artifacts**Metrics**

[branch_direct_ratio](#)
[branch_indirect_ratio](#)
[branch_return_ratio](#)
[branch_misprediction_ratio](#) in [Branch_Effectiveness](#)
[branch_misprediction_ratio](#) in [Miss_Ratio](#)

Metric groups

[Branch_Effectiveness](#)
[Miss_Ratio](#)

Functional groups

[Retired](#)

0x0022 BR_MIS_PRED_RETIRE, Branch instruction architecturally executed, mispredicted, event

Counts branches counted by BR_RETIRE which were mispredicted and caused a pipeline flush.

Related telemetry artifacts**Metrics**

[branch_mpki](#) in [Branch_Effectiveness](#)
[branch_mpki](#) in [MPKI](#)
[branch_misprediction_ratio](#) in [Branch_Effectiveness](#)
[branch_misprediction_ratio](#) in [Miss_Ratio](#)

Metric groups

[Branch_Effectiveness](#)
[MPKI](#)
[Miss_Ratio](#)

Functional groups

[Retired](#)

0x003A OP_RETIRE, Micro-operation architecturally executed, event

Counts micro-operations that are architecturally executed. This is a count of number of micro-operations retired from the commit queue in a single cycle.

Related telemetry artifacts**Metrics**

[retiring](#)
[bad_speculation](#)

Metric groups

[Topdown_L1](#)

Functional groups

[Retired](#)

0x8108 BR_IMMED_TAKEN_RETIRE, Branch instruction architecturally executed, immediate, taken, event

Counts architecturally executed direct branches that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x810c BR_INDNR_TAKEN_RETIRE, Branch instruction architecturally executed, indirect excluding procedure return, taken, event

Counts architecturally executed indirect branches excluding procedure returns that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x8110 BR_IMMED_PRED_RETIRE, Branch instruction architecturally executed, predicted immediate, event

Counts architecturally executed direct branches that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x8111 BR_IMMED_MIS_PRED_RETIRE, Branch instruction architecturally executed, mispredicted immediate, event

Counts architecturally executed direct branches that were mispredicted and caused a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x8112 BR_IND_PRED_RETIRE, Branch instruction architecturally executed, predicted indirect, event

Counts architecturally executed indirect branches including procedure returns that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x8113 BR_IND_MIS_PRED_RETIRE, Branch instruction architecturally executed, mispredicted indirect, event

Counts architecturally executed indirect branches including procedure returns that were mispredicted and caused a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x8114 BR_RETURN_PRED_RETIRE, Branch instruction architecturally executed, predicted procedure return, event

Counts architecturally executed procedure returns that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x8115 BR_RETURN_MIS_PRED_RETIRE, Branch instruction architecturally executed, mispredicted procedure return, event

Counts architecturally executed procedure returns that were mispredicted and caused a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Retired](#)

0x8116 BR_INDNR_PRED_RETIRE, Branch instruction architecturally executed, predicted indirect excluding procedure return, event

Counts architecturally executed indirect branches excluding procedure returns that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Retired](#)**0x8117 BR_INDNR_MIS_PRED_RETIRE, Branch instruction architecturally executed, mispredicted indirect excluding procedure return, event**

Counts architecturally executed indirect branches excluding procedure returns that were mispredicted and caused a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Retired](#)**0x811c BR_PRED_RETIRE, Branch instruction architecturally executed, predicted branch, event**

Counts branch instructions counted by BR_RETIRE which were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[Retired](#)**0x811d BR_IND_RETIRE, Instruction architecturally executed, indirect branch, event**

Counts architecturally executed indirect branches including procedure returns.

Related telemetry artifacts**Metrics**[branch_indirect_ratio](#)**Metric groups**[Branch_Effectiveness](#)**Functional groups**[Retired](#)

6.11 SPE (SPE) events for Neoverse N3

SPE related events.

Summary of events in SPE:

- Total implemented Common events: 10
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0

- ETE Only events : 0

Table 6-11: SPE events summary

Code	Mnemonic	Name	Description
0x4000	SAMPLE_POP	Statistical Profiling sample population	Counts statistical profiling sample population, the count of all operations that could be sampled...
0x4001	SAMPLE_FEED	Statistical Profiling sample taken	Counts statistical profiling samples taken for sampling.
0x4002	SAMPLE_FILTRATE	Statistical Profiling sample taken and not removed by filtering	Counts statistical profiling samples taken which are not removed by filtering.
0x4003	SAMPLE_COLLISION	Statistical Profiling sample collided with previous sample	Counts statistical profiling samples that have collided with a previous sample and so therefore...
0x812A	SAMPLE_FEED_BR	Statistical Profiling sample taken, branch	Counts statistical profiling samples taken which are branches.
0x812B	SAMPLE_FEED_LD	Statistical Profiling sample taken, load	Counts statistical profiling samples taken which are loads or load atomic operations.
0x812C	SAMPLE_FEED_ST	Statistical Profiling sample taken, store	Counts statistical profiling samples taken which are stores or store atomic operations.
0x812D	SAMPLE_FEED_OP	Statistical Profiling sample taken, matching operation type	Counts statistical profiling samples taken which are matching any operation type filters supported.
0x812E	SAMPLE_FEED_EVENT	Statistical Profiling sample taken, matching events	Counts statistical profiling samples taken which are matching event packet filter constraints.
0x812F	SAMPLE_FEED_LAT	Statistical Profiling sample taken, exceeding minimum latency	Counts statistical profiling samples taken which are exceeding minimum latency set by operation...

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x4000 SAMPLE_POP, Statistical Profiling sample population, event

Counts statistical profiling sample population, the count of all operations that could be sampled but may or may not be chosen for sampling.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[SPE](#)

0x4001 SAMPLE_FEED, Statistical Profiling sample taken, event

Counts statistical profiling samples taken for sampling.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[SPE](#)

0x4002 SAMPLE_FILTRATE, Statistical Profiling sample taken and not removed by filtering, event

Counts statistical profiling samples taken which are not removed by filtering.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[SPE](#)

0x4003 SAMPLE_COLLISION, Statistical Profiling sample collided with previous sample, event

Counts statistical profiling samples that have collided with a previous sample and so therefore not taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[SPE](#)

0x812A SAMPLE_FEED_BR, Statistical Profiling sample taken, branch, event

Counts statistical profiling samples taken which are branches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[SPE](#)

0x812B SAMPLE_FEED_LD, Statistical Profiling sample taken, load, event

Counts statistical profiling samples taken which are loads or load atomic operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[SPE](#)

0x812C SAMPLE_FEED_ST, Statistical Profiling sample taken, store, event

Counts statistical profiling samples taken which are stores or store atomic operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[SPE](#)**0x812D SAMPLE_FEED_OP, Statistical Profiling sample taken, matching operation type, event**

Counts statistical profiling samples taken which are matching any operation type filters supported.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[SPE](#)**0x812E SAMPLE_FEED_EVENT, Statistical Profiling sample taken, matching events, event**

Counts statistical profiling samples taken which are matching event packet filter constraints.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[SPE](#)**0x812F SAMPLE_FEED_LAT, Statistical Profiling sample taken, exceeding minimum latency, event**

Counts statistical profiling samples taken which are exceeding minimum latency set by operation latency filter constraints.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[SPE](#)

6.12 Spec_Operation (SPEC OPERATION) events for Neoverse N3

Speculatively executed operations related events.

Summary of events in Spec_Operation:

- Total implemented Common events: 22
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-12: Spec_Operation events summary

Code	Mnemonic	Name	Description
0x0010	BR_MIS_PRED	Branch instruction speculatively executed, mispredicted or not predicted	Counts branches which are speculatively executed and mispredicted.
0x0012	BR_PRED	Predictable branch instruction speculatively executed	Counts all speculatively executed branches.
0x001B	INST_SPEC	Operation speculatively executed	Counts operations that have been speculatively executed.
0x003B	OP_SPEC	Micro-operation speculatively executed	Counts micro-operations speculatively executed. This is the count of the number of...
0x006E	STREX_FAIL_SPEC	Exclusive operation speculatively executed, Store-Exclusive fail	Counts store-exclusive operations that have been speculatively executed and have not successfully...
0x006F	STREX_SPEC	Exclusive operation speculatively executed, Store-Exclusive	Counts store-exclusive operations that have been speculatively executed.
0x0070	LD_SPEC	Operation speculatively executed, load	Counts speculatively executed load operations including Single Instruction Multiple Data (SIMD)...
0x0071	ST_SPEC	Operation speculatively executed, store	Counts speculatively executed store operations including Single Instruction Multiple Data (SIMD)...
0x0073	DP_SPEC	Operation speculatively executed, integer data processing	Counts speculatively executed logical or arithmetic instructions such as MOV/MVN operations.
0x0074	ASE_SPEC	Operation speculatively executed, Advanced SIMD	Counts speculatively executed Advanced SIMD operations excluding load, store and move...
0x0075	VFP_SPEC	Operation speculatively executed, scalar floating-point	Counts speculatively executed floating point operations. This event does not count operations...
0x0076	PC_WRITE_SPEC	Operation speculatively executed, Software change of the PC	Counts speculatively executed operations which cause software changes of the PC. Those operations...
0x0077	CRYPTO_SPEC	Operation speculatively executed, Cryptographic instruction	Counts speculatively executed cryptographic operations except for PMULL and VMULL operations.
0x007C	ISB_SPEC	Barrier speculatively executed, ISB	Counts ISB operations that are executed.
0x007D	DSB_SPEC	Barrier speculatively executed, DSB	Counts DSB operations that are speculatively issued to Load/Store unit in the CPU.
0x007E	DMB_SPEC	Barrier speculatively executed, DMB	Counts DMB operations that are speculatively issued to the Load/Store unit in the CPU. This event...
0x0090	RC_LD_SPEC	Release consistency operation speculatively executed, Load-Acquire	Counts any load acquire operations that are speculatively executed. For example: LDAR, LDARH, LDARB
0x0091	RC_ST_SPEC	Release consistency operation speculatively executed, Store-Release	Counts any store release operations that are speculatively executed. For example: STLR, STLRH, STLRB
0x8005	ASE_INST_SPEC	Operation speculatively executed, Advanced SIMD	Counts speculatively executed Advanced SIMD operations.
0x8171	CAS_NEAR_PASS	Atomic memory Operation speculatively executed, Compare and Swap pass	This event counts compare and swap instructions that executed locally to the PE and updated the...
0x8172	CAS_NEAR_SPEC	Atomic memory Operation speculatively executed, Compare and Swap near	This event counts compare and swap instructions that executed locally to the PE.
0x8173	CAS_FAR_SPEC	Atomic memory Operation speculatively executed, Compare and Swap far	This event counts compare and swap instructions that did not execute locally to the PE.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0010 BR_MIS_PRED, Branch instruction speculatively executed, mispredicted or not predicted, event

Counts branches which are speculatively executed and mispredicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

0x0012 BR_PRED, Predictable branch instruction speculatively executed, event

Counts all speculatively executed branches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

0x001B INST_SPEC, Operation speculatively executed, event

Counts operations that have been speculatively executed.

Related telemetry artifacts**Metrics**

[load_percentage](#)
[store_percentage](#)
[integer_dp_percentage](#)
[simd_percentage](#)
[scalar_fp_percentage](#)
[barrier_percentage](#)
[branch_percentage](#)
[crypto_percentage](#)
[sve_all_percentage](#)
[sve_predicate_percentage](#)
[fp16_percentage](#)
[fp32_percentage](#)
[fp64_percentage](#)

Metric groups

[FP_Precision_Mix](#)
[Operation_Mix](#)
[SVE_Effectiveness](#)

Functional groups

[Spec_Operation](#)

0x003B OP_SPEC, Micro-operation speculatively executed, event

Counts micro-operations speculatively executed. This is the count of the number of micro-operations dispatched in a cycle.

Related telemetry artifacts**Metrics**

[retiring](#)
[bad_speculation](#)

Metric groups

[Topdown_L1](#)

Functional groups

[Spec_Operation](#)

0x006E STREX_FAIL_SPEC, Exclusive operation speculatively executed, Store-Exclusive fail, event

Counts store-exclusive operations that have been speculatively executed and have not successfully completed the store operation.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

0x006F STREX_SPEC, Exclusive operation speculatively executed, Store-Exclusive, event

Counts store-exclusive operations that have been speculatively executed.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

0x0070 LD_SPEC, Operation speculatively executed, load, event

Counts speculatively executed load operations including Single Instruction Multiple Data (SIMD) load operations.

Related telemetry artifacts**Metrics**

[load_percentage](#)

Metric groups

[Operation_Mix](#)

Functional groups

[Spec_Operation](#)

0x0071 ST_SPEC, Operation speculatively executed, store, event

Counts speculatively executed store operations including Single Instruction Multiple Data (SIMD) store operations.

Related telemetry artifacts**Metrics**[store_percentage](#)**Metric groups**[Operation_Mix](#)**Functional groups**[Spec_Operation](#)**0x0073 DP_SPEC, Operation speculatively executed, integer data processing, event**

Counts speculatively executed logical or arithmetic instructions such as MOV/MVN operations.

Related telemetry artifacts**Metrics**[integer_dp_percentage](#)**Metric groups**[Operation_Mix](#)**Functional groups**[Spec_Operation](#)**0x0074 ASE_SPEC, Operation speculatively executed, Advanced SIMD, event**

Counts speculatively executed Advanced SIMD operations excluding load, store and move micro-operations that move data to or from SIMD (vector) registers.

Related telemetry artifacts**Metrics**[simd_percentage](#)**Metric groups**[Operation_Mix](#)**Functional groups**[Spec_Operation](#)**0x0075 VFP_SPEC, Operation speculatively executed, scalar floating-point, event**

Counts speculatively executed floating point operations. This event does not count operations that move data to or from floating point (vector) registers.

Related telemetry artifacts**Metrics**[scalar_fp_percentage](#)

Metric groups[Operation_Mix](#)**Functional groups**[Spec_Operation](#)**0x0076 PC_WRITE_SPEC, Operation speculatively executed, Software change of the PC, event**

Counts speculatively executed operations which cause software changes of the PC. Those operations include all taken branch operations.

Related telemetry artifacts**Metrics**[branch_percentage](#)**Metric groups**[Operation_Mix](#)**Functional groups**[Spec_Operation](#)**0x0077 CRYPTO_SPEC, Operation speculatively executed, Cryptographic instruction, event**

Counts speculatively executed cryptographic operations except for PMULL and VMULL operations.

Related telemetry artifacts**Metrics**[crypto_percentage](#)**Metric groups**[Operation_Mix](#)**Functional groups**[Spec_Operation](#)**0x007c ISB_SPEC, Barrier speculatively executed, ISB, event**

Counts ISB operations that are executed.

Related telemetry artifacts**Metrics**[barrier_percentage](#)**Metric groups**[Operation_Mix](#)**Functional groups**[Spec_Operation](#)**0x007d DSB_SPEC, Barrier speculatively executed, DSB, event**

Counts DSB operations that are speculatively issued to Load/Store unit in the CPU.

Related telemetry artifacts**Metrics**

[integer_dp_percentage](#)
[barrier_percentage](#)

Metric groups

[Operation_Mix](#)

Functional groups

[Spec_Operation](#)

0x007E DMB_SPEC, Barrier speculatively executed, DMB, event

Counts DMB operations that are speculatively issued to the Load/Store unit in the CPU. This event does not count implied barriers from load acquire/store release operations.

Related telemetry artifacts**Metrics**

[barrier_percentage](#)

Metric groups

[Operation_Mix](#)

Functional groups

[Spec_Operation](#)

0x0090 RC_LD_SPEC, Release consistency operation speculatively executed, Load-Acquire, event

Counts any load acquire operations that are speculatively executed. For example: LDAR, LDARH, LDARB

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

0x0091 RC_ST_SPEC, Release consistency operation speculatively executed, Store-Release, event

Counts any store release operations that are speculatively executed. For example: STLR, STLRH, STLRB

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

0x8005 ASE_INST_SPEC, Operation speculatively executed, Advanced SIMD, event

Counts speculatively executed Advanced SIMD operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

0x8171 CAS_NEAR_PASS, Atomic memory Operation speculatively executed, Compare and Swap pass, event

This event counts compare and swap instructions that executed locally to the PE and updated the location accessed.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

0x8172 CAS_NEAR_SPEC, Atomic memory Operation speculatively executed, Compare and Swap near, event

This event counts compare and swap instructions that executed locally to the PE.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

0x8173 CAS_FAR_SPEC, Atomic memory Operation speculatively executed, Compare and Swap far, event

This event counts compare and swap instructions that did not execute locally to the PE.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Spec_Operation](#)

6.13 FP_Operation (FP OPERATION) events for Neoverse N3

Speculatively executed floating-point events.

Summary of events in FP_Operation:

- Total implemented Common events: 5
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-13: FP_Operation events summary

Code	Mnemonic	Name	Description
0x8014	FP_HP_SPEC	Floating-point operation speculatively executed, half precision	Counts speculatively executed half precision floating point operations.
0x8018	FP_SP_SPEC	Floating-point operation speculatively executed, single precision	Counts speculatively executed single precision floating point operations.
0x801C	FP_DP_SPEC	Floating-point operation speculatively executed, double precision	Counts speculatively executed double precision floating point operations.
0x80C0	FP_SCALE_OPS_SPEC	Scalable floating-point element ALU operations speculatively executed	Counts speculatively executed scalable single precision floating point operations.
0x80C1	FP_FIXED_OPS_SPEC	Non-scalable floating-point element ALU operations speculatively executed	Counts speculatively executed non-scalable single precision floating point operations.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x8014 FP_HP_SPEC, Floating-point operation speculatively executed, half precision, event

Counts speculatively executed half precision floating point operations.

Related telemetry artifacts

Metrics

[fp16_percentage](#)

Metric groups

[FP_Precision_Mix](#)

Functional groups

[FP_Operation](#)

0x8018 FP_SP_SPEC, Floating-point operation speculatively executed, single precision, event

Counts speculatively executed single precision floating point operations.

Related telemetry artifacts**Metrics**[fp32_percentage](#)**Metric groups**[FP_Precision_Mix](#)**Functional groups**[FP_Operation](#)**0x801c FP_DP_SPEC, Floating-point operation speculatively executed, double precision, event**

Counts speculatively executed double precision floating point operations.

Related telemetry artifacts**Metrics**[fp64_percentage](#)**Metric groups**[FP_Precision_Mix](#)**Functional groups**[FP_Operation](#)**0x80c0 FP_SCALE_OPS_SPEC, Scalable floating-point element ALU operations speculatively executed, event**

Counts speculatively executed scalable single precision floating point operations.

Related telemetry artifacts**Metrics**[sve_fp_ops_per_cycle](#)[fp_ops_per_cycle](#)**Metric groups**[FP_Arithmetic_Intensity](#)**Functional groups**[FP_Operation](#)**0x80c1 FP_FIXED_OPS_SPEC, Non-scalable floating-point element ALU operations speculatively executed, event**

Counts speculatively executed non-scalable single precision floating point operations.

Related telemetry artifacts**Metrics**[nonsve_fp_ops_per_cycle](#)[fp_ops_per_cycle](#)

Metric groups

FP_Arithmetic_Intensity

Functional groups

FP_Operation

6.14 Stall (STALL) events for Neoverse N3

Stall related events.

Summary of events in Stall:

- Total implemented Common events: 21
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-14: Stall events summary

Code	Mnemonic	Name	Description
0x0023	STALL_FRONTEND	No operation sent for execution due to the frontend	Counts cycles when frontend could not send any micro-operations to the rename stage because of...
0x0024	STALL_BACKEND	No operation sent for execution due to the backend	Counts cycles whenever the rename unit is unable to send any micro-operations to the backend of...
0x003C	STALL	No operation sent for execution	Counts cycles when no operations are sent to the rename unit from the frontend or from the rename...
0x003D	STALL_SLOT_BACKEND	No operation sent for execution on a Slot due to the backend	Counts slots per cycle in which no operations are sent from the rename unit to the backend due to...
0x003E	STALL_SLOT_FRONTEND	No operation sent for execution on a Slot due to the frontend	Counts slots per cycle in which no operations are sent to the rename unit from the frontend due...
0x003F	STALL_SLOT	No operation sent for execution on a Slot	Counts slots per cycle in which no operations are sent to the rename unit from the frontend or...
0x4005	STALL_BACKEND_MEM	Memory stall cycles	Counts cycles when the backend is stalled because there is a pending demand load request in...
0x8158	STALL_FRONTEND_MEMBOUND	Frontend stall cycles, memory bound	Counts cycles when the frontend could not send any micro-operations to the rename stage due to...
0x8159	STALL_FRONTEND_L1I	Frontend stall cycles, level 1 instruction cache	Counts cycles when the frontend is stalled because there is an instruction fetch request pending...
0x815B	STALL_FRONTEND_MEM	Frontend stall cycles, last level PE cache or memory	Counts cycles when the frontend is stalled because there is an instruction fetch request pending...
0x815C	STALL_FRONTEND_TLB	Frontend stall cycles, TLB	Counts when the frontend is stalled on any TLB misses being handled. This event also counts the...
0x8160	STALL_FRONTEND_CPUBOUND	Frontend stall cycles, processor bound	Counts cycles when the frontend could not send any micro-operations to the rename stage due to...
0x8161	STALL_FRONTEND_FLOW	Frontend stall cycles, flow control	Counts cycles when the frontend could not send any micro-operations to the rename stage due to...

Code	Mnemonic	Name	Description
0x8162	STALL_FRONTEND_FLUSH	Frontend stall cycles, flush recovery	Counts cycles when the frontend could not send any micro-operations to the rename stage as the...
0x8164	STALL_BACKEND_MEMBOUND	Backend stall cycles, memory bound	Counts cycles when the backend could not accept any micro-operations due to resource constraints...
0x8165	STALL_BACKEND_L1D	Backend stall cycles, level 1 data cache	Counts cycles when the backend is stalled because there is a pending demand load request in...
0x8167	STALL_BACKEND_TLB	Backend stall cycles, TLB	Counts cycles when the backend is stalled on any demand TLB misses being handled.
0x8168	STALL_BACKEND_ST	Backend stall cycles, store	Counts cycles when the backend is stalled and there is a store that has not reached the...
0x816A	STALL_BACKEND_CPUBOUND	Backend stall cycles, processor bound	Counts cycles when the backend could not accept any micro-operations due to any resource...
0x816B	STALL_BACKEND_BUSY	Backend stall cycles, backend busy	Counts cycles when the backend could not accept any micro-operations because the issue queues are...
0x816D	STALL_BACKEND_RENAME	Backend stall cycles, rename full	Counts cycles when backend is stalled even when operations are available from the frontend but at...

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0023 STALL_FRONTEND, No operation sent for execution due to the frontend, event

Counts cycles when frontend could not send any micro-operations to the rename stage because of frontend resource stalls caused by fetch memory latency or branch prediction flow stalls. STALL_FRONTEND_SLOTS counts SLOTS during the cycle when this event counts.

Related telemetry artifacts

Metrics

[frontend_stalled_cycles](#)
[frontend_core_bound](#)
[frontend_mem_bound](#)

Metric groups

[Cycle_Accounting](#)
[Topdown_Frontend](#)

Functional groups

[Stall](#)

0x0024 STALL_BACKEND, No operation sent for execution due to the backend, event

Counts cycles whenever the rename unit is unable to send any micro-operations to the backend of the pipeline because of backend resource constraints. Backend resource constraints can include issue stage fullness, execution stage fullness, or other internal pipeline resource fullness. All the backend slots were empty during the cycle when this event counts.

Related telemetry artifacts

Metrics

[backend_stalled_cycles](#)

backend_core_bound
backend_mem_bound
backend_busy_bound

Metric groups

Cycle_Accounting
Topdown_Backend

Functional groups

Stall

0x003C STALL, No operation sent for execution, event

Counts cycles when no operations are sent to the rename unit from the frontend or from the rename unit to the backend for any reason (either frontend or backend stall). This event is the sum of STALL_FRONTEND and STALL_BACKEND

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

Stall

0x003D STALL_SLOT_BACKEND, No operation sent for execution on a Slot due to the backend, event

Counts slots per cycle in which no operations are sent from the rename unit to the backend due to backend resource constraints. STALL_BACKEND counts during the cycle when STALL_SLOT_BACKEND counts at least 1.

Related telemetry artifacts**Metrics**

backend_bound

Metric groups

Topdown_L1

Functional groups

Stall

0x003E STALL_SLOT_FRONTEND, No operation sent for execution on a Slot due to the frontend, event

Counts slots per cycle in which no operations are sent to the rename unit from the frontend due to frontend resource constraints.

Related telemetry artifacts**Metrics**

frontend_bound

Metric groups

Topdown_L1

Functional groups

[Stall](#)

0x003F STALL_SLOT, No operation sent for execution on a Slot, event

Counts slots per cycle in which no operations are sent to the rename unit from the frontend or from the rename unit to the backend for any reason (either frontend or backend stall). STALL_SLOT is the sum of STALL_SLOT_FRONTEND and STALL_SLOT_BACKEND.

Related telemetry artifacts

Metrics

[retiring](#)

[bad_speculation](#)

Metric groups

[Topdown_L1](#)

Functional groups

[Stall](#)

0x4005 STALL_BACKEND_MEM, Memory stall cycles, event

Counts cycles when the backend is stalled because there is a pending demand load request in progress in the last level core cache.

Related telemetry artifacts

Metrics

[backend_mem_cache_bound](#)

[backend_cache_l1d_bound](#)

[backend_cache_l2d_bound](#)

Metric groups

[Topdown_Backend](#)

Functional groups

[Stall](#)

0x8158 STALL_FRONTEND_MEMBOUND, Frontend stall cycles, memory bound, event

Counts cycles when the frontend could not send any micro-operations to the rename stage due to resource constraints in the memory resources.

Related telemetry artifacts

Metrics

[frontend_mem_bound](#)

[frontend_mem_cache_bound](#)

[frontend_mem_tlb_bound](#)

Metric groups

[Topdown_Frontend](#)

Functional groups[Stall](#)**0x8159 STALL_FRONTEND_L1I, Frontend stall cycles, level 1 instruction cache, event**

Counts cycles when the frontend is stalled because there is an instruction fetch request pending in the level 1 instruction cache.

Related telemetry artifacts**Metrics**[frontend_mem_cache_bound](#)[frontend_cache_l1i_bound](#)[frontend_cache_l2i_bound](#)**Metric groups**[Topdown_Frontend](#)**Functional groups**[Stall](#)**0x815B STALL_FRONTEND_MEM, Frontend stall cycles, last level PE cache or memory, event**

Counts cycles when the frontend is stalled because there is an instruction fetch request pending in the last level core cache.

Related telemetry artifacts**Metrics**[frontend_mem_cache_bound](#)[frontend_cache_l1i_bound](#)[frontend_cache_l2i_bound](#)**Metric groups**[Topdown_Frontend](#)**Functional groups**[Stall](#)**0x815C STALL_FRONTEND_TLB, Frontend stall cycles, TLB, event**

Counts when the frontend is stalled on any TLB misses being handled. This event also counts the TLB accesses made by hardware prefetches.

Related telemetry artifacts**Metrics**[frontend_mem_tlb_bound](#)**Metric groups**[Topdown_Frontend](#)**Functional groups**[Stall](#)

0x8160 STALL_FRONTEND_CPUBOUND, Frontend stall cycles, processor bound, event

Counts cycles when the frontend could not send any micro-operations to the rename stage due to resource constraints in the CPU resources excluding memory resources.

Related telemetry artifacts**Metrics**

[frontend_core_bound](#)
[frontend_core_flush_bound](#)
[frontend_core_flow_bound](#)

Metric groups

[Topdown_Frontend](#)

Functional groups

[Stall](#)

0x8161 STALL_FRONTEND_FLOW, Frontend stall cycles, flow control, event

Counts cycles when the frontend could not send any micro-operations to the rename stage due to resource constraints in the branch prediction unit.

Related telemetry artifacts**Metrics**

[frontend_core_flow_bound](#)

Metric groups

[Topdown_Frontend](#)

Functional groups

[Stall](#)

0x8162 STALL_FRONTEND_FLUSH, Frontend stall cycles, flush recovery, event

Counts cycles when the frontend could not send any micro-operations to the rename stage as the frontend is recovering from a machine flush or resteer. Example scenarios that cause a flush include branch mispredictions, taken exceptions, micro-architectural flush etc.

Related telemetry artifacts**Metrics**

[frontend_bound](#)
[bad_speculation](#)
[frontend_core_flush_bound](#)

Metric groups

[Topdown_Frontend](#)
[Topdown_L1](#)

Functional groups

[Stall](#)

0x8164 STALL_BACKEND_MEMBOUND, Backend stall cycles, memory bound, event

Counts cycles when the backend could not accept any micro-operations due to resource constraints in the memory resources.

Related telemetry artifacts**Metrics**

[backend_mem_bound](#)
[backend_mem_cache_bound](#)
[backend_mem_tlb_bound](#)
[backend_mem_store_bound](#)

Metric groups

[Topdown_Backend](#)

Functional groups

[Stall](#)

0x8165 STALL_BACKEND_L1D, Backend stall cycles, level 1 data cache, event

Counts cycles when the backend is stalled because there is a pending demand load request in progress in the level 1 data cache.

Related telemetry artifacts**Metrics**

[backend_mem_cache_bound](#)
[backend_cache_l1d_bound](#)
[backend_cache_l2d_bound](#)

Metric groups

[Topdown_Backend](#)

Functional groups

[Stall](#)

0x8167 STALL_BACKEND_TLB, Backend stall cycles, TLB, event

Counts cycles when the backend is stalled on any demand TLB misses being handled.

Related telemetry artifacts**Metrics**

[backend_mem_tlb_bound](#)

Metric groups

[Topdown_Backend](#)

Functional groups

[Stall](#)

0x8168 STALL_BACKEND_ST, Backend stall cycles, store, event

Counts cycles when the backend is stalled and there is a store that has not reached the pre-commit stage.

Related telemetry artifacts**Metrics**[backend_mem_store_bound](#)**Metric groups**[Topdown_Backend](#)**Functional groups**[Stall](#)**0x816A STALL_BACKEND_CPUBOUND, Backend stall cycles, processor bound, event**

Counts cycles when the backend could not accept any micro-operations due to any resource constraints in the CPU excluding memory resources.

Related telemetry artifacts**Metrics**[backend_core_bound](#)[backend_core_rename_bound](#)**Metric groups**[Topdown_Backend](#)**Functional groups**[Stall](#)**0x816B STALL_BACKEND_BUSY, Backend stall cycles, backend busy, event**

Counts cycles when the backend could not accept any micro-operations because the issue queues are full to take any operations for execution.

Related telemetry artifacts**Metrics**[backend_busy_bound](#)**Metric groups**[Topdown_Backend](#)**Functional groups**[Stall](#)**0x816D STALL_BACKEND_RENAME, Backend stall cycles, rename full, event**

Counts cycles when backend is stalled even when operations are available from the frontend but at least one is not ready to be sent to the backend because no rename register is available.

Related telemetry artifacts**Metrics**[backend_core_rename_bound](#)**Metric groups**[Topdown_Backend](#)

Functional groups[Stall](#)

6.15 General (GENERAL) events for Neoverse N3

General CPU related events.

Summary of events in General:

- Total implemented Common events: 2
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-15: General events summary

Code	Mnemonic	Name	Description
0x0011	CPU_CYCLES	Cycle	Counts CPU clock cycles (not timer cycles). The clock measured by this event is defined as the...
0x4004	CNT_CYCLES	Constant frequency cycles	Increments at a constant frequency equal to the rate of increment of the System Counter, CNTPCT_ELO.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0011 CPU_CYCLES, Cycle, event

Counts CPU clock cycles (not timer cycles). The clock measured by this event is defined as the physical clock driving the CPU logic.

Related telemetry artifacts

Metrics

[frontend_stalled_cycles](#)
[backend_stalled_cycles](#)
[frontend_bound](#)
[backend_bound](#)
[retiring](#)
[bad_speculation](#)
[ipc](#)

Metric groups

[Cycle_Accounting](#)
[General](#)
[Topdown_L1](#)

Functional groups

[General](#)

0x4004 CNT_CYCLES, Constant frequency cycles, event

Increments at a constant frequency equal to the rate of increment of the System Counter, CNTPCT_ELO.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

General

6.16 TLB (TLB) events for Neoverse N3

TLB and MMU related events.

Summary of events in TLB:

- Total implemented Common events: 18
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-16: TLB events summary

Code	Mnemonic	Name	Description
0x0002	L1I_TLB_REFILL	Level 1 instruction TLB refill	Counts level 1 instruction TLB refills from any Instruction fetch. If there are multiple misses...
0x0005	L1D_TLB_REFILL	Level 1 data TLB refill	Counts level 1 data TLB accesses that resulted in TLB refills. If there are multiple misses in...
0x0025	L1D_TLB	Level 1 data TLB access	Counts level 1 data TLB accesses caused by any memory load or store operation. Note that load or...
0x0026	L1I_TLB	Level 1 instruction TLB access	Counts level 1 instruction TLB accesses, whether the access hits or misses in the TLB. This event...
0x002D	L2D_TLB_REFILL	Level 2 data TLB refill	Counts level 2 TLB refills caused by memory operations from both data and instruction fetch,...
0x002F	L2D_TLB	Level 2 data TLB access	Counts level 2 TLB accesses except those caused by TLB maintenance operations.
0x0034	DTLB_WALK	Data TLB access with at least one translation table walk	Counts number of demand data translation table walks caused by a miss in the L2 TLB and...
0x0035	ITLB_WALK	Instruction TLB access with at least one translation table walk	Counts number of instruction translation table walks caused by a miss in the L2 TLB and...
0x8128	DTLB_WALK_PERCYC	Event in progress, DTLB WALK	Counts the number of data translation table walks in progress per cycle.
0x8129	ITLB_WALK_PERCYC	Event in progress, ITLB WALK	Counts the number of instruction translation table walks in progress per cycle.
0x8134	DTLB_HWUPD	Data TLB hardware update of translation table	Counts number of memory accesses triggered by a data translation table walk and performing an...

Code	Mnemonic	Name	Description
0x8135	ITLB_HWUPD	Instruction TLB hardware update of translation table	Counts number of memory accesses triggered by an instruction translation table walk and...
0x8136	DTLB_STEP	Data TLB translation table walk, step	Counts number of memory accesses triggered by a demand data translation table walk and performing...
0x8137	ITLB_STEP	Instruction TLB translation table walk, step	Counts number of memory accesses triggered by an instruction translation table walk and...
0x8138	DTLB_WALK_LARGE	Data TLB large page translation table walk	Counts number of demand data translation table walks caused by a miss in the L2 TLB and yielding...
0x8139	ITLB_WALK_LARGE	Instruction TLB large page translation table walk	Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding...
0x813A	DTLB_WALK_SMALL	Data TLB small page translation table walk	Counts number of data translation table walks caused by a miss in the L2 TLB and yielding a small...
0x813B	ITLB_WALK_SMALL	Instruction TLB small page translation table walk	Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding...

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x0002 L1I_TLB_REFILL, Level 1 instruction TLB refill, event

Counts level 1 instruction TLB refills from any Instruction fetch. If there are multiple misses in the TLB that are resolved by the refill, then this event only counts once. This event will not count if the translation table walk results in a fault (such as a translation or access fault), since there is no new translation created for the TLB.

Related telemetry artifacts

Metrics

[l1i_tlb_mpki](#) in [ITLB_Effectiveness](#)
[l1i_tlb_mpki](#) in [MPKI](#)
[l1i_tlb_miss_ratio](#) in [ITLB_Effectiveness](#)
[l1i_tlb_miss_ratio](#) in [Miss_Ratio](#)

Metric groups

[ITLB_Effectiveness](#)
[MPKI](#)
[Miss_Ratio](#)

Functional groups

[TLB](#)

0x0005 L1D_TLB_REFILL, Level 1 data TLB refill, event

Counts level 1 data TLB accesses that resulted in TLB refills. If there are multiple misses in the TLB that are resolved by the refill, then this event only counts once. This event counts for refills caused by preload instructions or hardware prefetch accesses. This event counts regardless of whether the miss hits in L2 or results in a translation table walk. This event will not count if the translation table walk results in a fault (such as a translation or access fault), since there is no new translation created for the TLB. This event will not count on an access from an AT(address translation) instruction.

Related telemetry artifacts

Metrics

- [l1d_tlb_mpki in DTLB_Effectiveness](#)
- [l1d_tlb_mpki in MPKI](#)
- [l1d_tlb_miss_ratio in DTLB_Effectiveness](#)
- [l1d_tlb_miss_ratio in Miss_Ratio](#)

Metric groups

- [DTLB_Effectiveness](#)
- [MPKI](#)
- [Miss_Ratio](#)

Functional groups

- [TLB](#)

0x0025 L1D_TLB, Level 1 data TLB access, event

Counts level 1 data TLB accesses caused by any memory load or store operation. Note that load or store instructions can be broken up into multiple memory operations. This event does not count TLB maintenance operations.

Related telemetry artifacts

Metrics

- [dtlb_walk_ratio in DTLB_Effectiveness](#)
- [dtlb_walk_ratio in Miss_Ratio](#)
- [l1d_tlb_miss_ratio in DTLB_Effectiveness](#)
- [l1d_tlb_miss_ratio in Miss_Ratio](#)

Metric groups

- [DTLB_Effectiveness](#)
- [Miss_Ratio](#)

Functional groups

- [TLB](#)

0x0026 L1I_TLB, Level 1 instruction TLB access, event

Counts level 1 instruction TLB accesses, whether the access hits or misses in the TLB. This event counts both demand accesses and prefetch or preload generated accesses.

Related telemetry artifacts

Metrics

- [itlb_walk_ratio in ITLB_Effectiveness](#)
- [itlb_walk_ratio in Miss_Ratio](#)
- [l1i_tlb_miss_ratio in ITLB_Effectiveness](#)
- [l1i_tlb_miss_ratio in Miss_Ratio](#)

Metric groups

- [ITLB_Effectiveness](#)

[Miss_Ratio](#)**Functional groups**[TLB](#)**0x002D L2D_TLB_REFILL, Level 2 data TLB refill, event**

Counts level 2 TLB refills caused by memory operations from both data and instruction fetch, except for those caused by TLB maintenance operations and hardware prefetches.

Related telemetry artifacts**Metrics**

[l2_tlb_mpki](#) in [DTLB_Effectiveness](#)
[l2_tlb_mpki](#) in [ITLB_Effectiveness](#)
[l2_tlb_mpki](#) in [MPKI](#)
[l2_tlb_miss_ratio](#) in [DTLB_Effectiveness](#)
[l2_tlb_miss_ratio](#) in [ITLB_Effectiveness](#)
[l2_tlb_miss_ratio](#) in [Miss_Ratio](#)

Metric groups

[DTLB_Effectiveness](#)
[ITLB_Effectiveness](#)
[MPKI](#)
[Miss_Ratio](#)

Functional groups[TLB](#)**0x002F L2D_TLB, Level 2 data TLB access, event**

Counts level 2 TLB accesses except those caused by TLB maintenance operations.

Related telemetry artifacts**Metrics**

[l2_tlb_miss_ratio](#) in [DTLB_Effectiveness](#)
[l2_tlb_miss_ratio](#) in [ITLB_Effectiveness](#)
[l2_tlb_miss_ratio](#) in [Miss_Ratio](#)

Metric groups

[DTLB_Effectiveness](#)
[ITLB_Effectiveness](#)
[Miss_Ratio](#)

Functional groups[TLB](#)**0x0034 DTLB_WALK, Data TLB access with at least one translation table walk, event**

Counts number of demand data translation table walks caused by a miss in the L2 TLB and performing at least one memory access. Translation table walks are counted even if the translation

ended up taking a translation fault for reasons different than EPD, EOPD and NFD. Note that partial translations that cause a translation table walk are also counted. Also note that this event counts walks triggered by software preloads, but not walks triggered by hardware prefetchers, and that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

Metrics

[dtlb_mpki in DTLB_Effectiveness](#)
[dtlb_mpki in MPKI](#)
[dtlb_walk_ratio in DTLB_Effectiveness](#)
[dtlb_walk_ratio in Miss_Ratio](#)

Metric groups

[DTLB_Effectiveness](#)
[MPKI](#)
[Miss_Ratio](#)

Functional groups

[TLB](#)

0x0035 ITLB_WALK, Instruction TLB access with at least one translation table walk, event

Counts number of instruction translation table walks caused by a miss in the L2 TLB and performing at least one memory access. Translation table walks are counted even if the translation ended up taking a translation fault for reasons different than EPD, EOPD and NFD. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

Metrics

[itlb_mpki in ITLB_Effectiveness](#)
[itlb_mpki in MPKI](#)
[itlb_walk_ratio in ITLB_Effectiveness](#)
[itlb_walk_ratio in Miss_Ratio](#)

Metric groups

[ITLB_Effectiveness](#)
[MPKI](#)
[Miss_Ratio](#)

Functional groups

[TLB](#)

0x8128 DTLB_WALK_PERCYC, Event in progress, DTLB WALK, event

Counts the number of data translation table walks in progress per cycle.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

0x8129 ITLB_WALK_PERCYC, Event in progress, ITLB WALK, event

Counts the number of instruction translation table walks in progress per cycle.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

0x8134 DTLB_HWUPD, Data TLB hardware update of translation table, event

Counts number of memory accesses triggered by a data translation table walk and performing an update of a translation table entry. Memory accesses are counted even if the translation ended up taking a translation fault for reasons different than EPD, EOPD and NFD. Note that this event counts accesses triggered by software preloads, but not accesses triggered by hardware prefetchers.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

0x8135 ITLB_HWUPD, Instruction TLB hardware update of translation table, event

Counts number of memory accesses triggered by an instruction translation table walk and performing an update of a translation table entry. Memory accesses are counted even if the translation ended up taking a translation fault for reasons different than EPD, EOPD and NFD.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

0x8136 DTLB_STEP, Data TLB translation table walk, step, event

Counts number of memory accesses triggered by a demand data translation table walk and performing a read of a translation table entry. Memory accesses are counted even if the translation ended up taking a translation fault for reasons different than EPD, EOPD and NFD. Note that this event counts accesses triggered by software preloads, but not accesses triggered by hardware prefetchers.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

0x8137 ITLB_STEP, Instruction TLB translation table walk, step, event

Counts number of memory accesses triggered by an instruction translation table walk and performing a read of a translation table entry. Memory accesses are counted even if the translation ended up taking a translation fault for reasons different than EPD, EOPD and NFD.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

0x8138 DTLB_WALK_LARGE, Data TLB large page translation table walk, event

Counts number of demand data translation table walks caused by a miss in the L2 TLB and yielding a large page. The set of large pages is defined as all pages with a final size higher than or equal to 2MB. Translation table walks that end up taking a translation fault are not counted, as the page size would be undefined in that case. If DTLB_WALK_BLOCK is implemented, then it is an alias for this event in this family. Note that partial translations that cause a translation table walk are also counted. Also note that this event counts walks triggered by software preloads, but not walks triggered by hardware prefetchers, and that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

0x8139 ITLB_WALK_LARGE, Instruction TLB large page translation table walk, event

Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding a large page. The set of large pages is defined as all pages with a final size higher than or equal to 2MB. Translation table walks that end up taking a translation fault are not counted, as the page size would be undefined in that case. In this family, this is equal to ITLB_WALK_BLOCK event. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

0x813A DTLB_WALK_SMALL, Data TLB small page translation table walk, event

Counts number of data translation table walks caused by a miss in the L2 TLB and yielding a small page. The set of small pages is defined as all pages with a final size lower than 2MB. Translation table walks that end up taking a translation fault are not counted, as the page size would be undefined in that case. If DTLB_WALK_PAGE event is implemented, then it is an alias for this event in this family. Note that partial translations that cause a translation table walk are also counted. Also note that this event counts walks triggered by software preloads, but not walks triggered by hardware prefetchers, and that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

0x813B ITLB_WALK_SMALL, Instruction TLB small page translation table walk, event

Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding a small page. The set of small pages is defined as all pages with a final size lower than 2MB. Translation table walks that end up taking a translation fault are not counted, as the page size would be undefined in that case. In this family, this is equal to ITLB_WALK_PAGE event. Note that partial translations that cause a translation table walk are also counted. Also note that this event does not count walks triggered by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TLB

6.17 SVE (SVE) events for Neoverse N3

SVE related events.

Summary of events in SVE:

- Total implemented Common events: 12
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-17: SVE events summary

Code	Mnemonic	Name	Description
0x8006	SVE_INST_SPEC	Operation speculatively executed, SVE, including load and store	Counts speculatively executed operations that are SVE operations.

Code	Mnemonic	Name	Description
0x8074	SVE_PRED_SPEC	Operation speculatively executed, SVE predicated	Counts speculatively executed predicated SVE operations.
0x8075	SVE_PRED_EMPTY_SPEC	Operation speculatively executed, SVE predicated with no active predicates	Counts speculatively executed predicated SVE operations with no active predicate elements.
0x8076	SVE_PRED_FULL_SPEC	Operation speculatively executed, SVE predicated with all active predicates	Counts speculatively executed predicated SVE operations with all predicate elements active.
0x8077	SVE_PRED_PARTIAL_SPEC	Operation speculatively executed, SVE predicated with partially active predicates	Counts speculatively executed predicated SVE operations with at least one but not all active...
0x8079	SVE_PRED_NOT_FULL_SPEC	SVE predicated operations speculatively executed with no active or partially active predicates	Counts speculatively executed predicated SVE operations with at least one non active predicate...
0x80BC	SVE_LDFF_SPEC	Operation speculatively executed, SVE first-fault load	Counts speculatively executed SVE first fault or non-fault load operations.
0x80BD	SVE_LDFF_FAULT_SPEC	Operation speculatively executed, SVE first-fault load which set FFR bit to 0b0	Counts speculatively executed SVE first fault or non-fault load operations that clear at least...
0x80E3	ASE_SVE_INT8_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 8-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...
0x80E7	ASE_SVE_INT16_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 16-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...
0x80EB	ASE_SVE_INT32_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 32-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...
0x80EF	ASE_SVE_INT64_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 64-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x8006 SVE_INST_SPEC, Operation speculatively executed, SVE, including load and store, event

Counts speculatively executed operations that are SVE operations.

Related telemetry artifacts

Metrics

[sve_all_percentage](#)

Metric groups

[Operation_Mix](#)

Functional groups

[SVE](#)

0x8074 SVE_PRED_SPEC, Operation speculatively executed, SVE predicated, event

Counts speculatively executed predicated SVE operations.

Related telemetry artifacts**Metrics**

[sve_predicate_percentage](#)
[sve_predicate_full_percentage](#)
[sve_predicate_partial_percentage](#)
[sve_predicate_empty_percentage](#)

Metric groups

[SVE_Effectiveness](#)

Functional groups

[SVE](#)

0x8075 SVE_PRED_EMPTY_SPEC, Operation speculatively executed, SVE predicated with no active predicates, event

Counts speculatively executed predicated SVE operations with no active predicate elements.

Related telemetry artifacts**Metrics**

[sve_predicate_empty_percentage](#)

Metric groups

[SVE_Effectiveness](#)

Functional groups

[SVE](#)

0x8076 SVE_PRED_FULL_SPEC, Operation speculatively executed, SVE predicated with all active predicates, event

Counts speculatively executed predicated SVE operations with all predicate elements active.

Related telemetry artifacts**Metrics**

[sve_predicate_full_percentage](#)

Metric groups

[SVE_Effectiveness](#)

Functional groups

[SVE](#)

0x8077 SVE_PRED_PARTIAL_SPEC, Operation speculatively executed, SVE predicated with partially active predicates, event

Counts speculatively executed predicated SVE operations with at least one but not all active predicate elements.

Related telemetry artifacts**Metrics**

[sve_predicate_partial_percentage](#)

Metric groups[SVE_Effectiveness](#)**Functional groups**[SVE](#)**0x8079 SVE_PRED_NOT_FULL_SPEC, SVE predicated operations speculatively executed with no active or partially active predicates, event**

Counts speculatively executed predicated SVE operations with at least one non active predicate elements.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[SVE](#)**0x80BC SVE_LDFF_SPEC, Operation speculatively executed, SVE first-fault load, event**

Counts speculatively executed SVE first fault or non-fault load operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[SVE](#)**0x80BD SVE_LDFF_FAULT_SPEC, Operation speculatively executed, SVE first-fault load which set FFR bit to 0b0, event**

Counts speculatively executed SVE first fault or non-fault load operations that clear at least one bit in the FFR.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[SVE](#)**0x80E3 ASE_SVE_INT8_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 8-bit, event**

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type an 8-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups[SVE](#)

0x80E7 ASE_SVE_INT16_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 16-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type a 16-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[SVE](#)

0x80EB ASE_SVE_INT32_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 32-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type a 32-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[SVE](#)

0x80EF ASE_SVE_INT64_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 64-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type a 64-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[SVE](#)

6.18 Non_PMU (NON_PMU) events for Neoverse N3

Non-PMU related events.

Summary of events in Non_PMU:

- Total implemented Common events: 2
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 2

Table 6-18: Non_PMU events summary

Code	Mnemonic	Name	Description
0x400D	PMU_OVFS	PMU overflow, counters accessible to EL1 and EL0	This event is generated each time an event causes a PMEVTCTNR<n>_EL1 counter overflow when...
0x400F	PMU_HOVFS	PMU overflow, counters reserved for use by EL2	This event is generated each time an event causes a PMEVTCTNR<n>_EL1 counter overflow when...

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x400D PMU_OVFS, PMU overflow, counters accessible to EL1 and EL0, event

This event is generated each time an event causes a PMEVTCTNR<n>_EL1 counter overflow when PMINTENSET_EL1[n] is set to 1, for each implemented PMU counter n in the range $0 \leq n < \text{UInt}(\text{MDCR_EL2.HPMN})$, and the Cycle Counter (n = 31).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Non_PMU](#)

0x400F PMU_HOVFS, PMU overflow, counters reserved for use by EL2, event

This event is generated each time an event causes a PMEVTCTNR<n>_EL1 counter overflow when PMINTENSET_EL1[n] is set to 1, for each implemented PMU counter n in the range $\text{UInt}(\text{MDCR_EL2.HPMN}) \leq n < \text{UInt}(\text{PMCR_ELO.N})$. This event is not transmitted to a PE Trace Unit while TRCFR_EL2.E2TRE == 0b0.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Non_PMU](#)

6.19 TRACE (TRACE) events for Neoverse N3

Trace related events.

Summary of events in TRACE:

- Total implemented Common events: 10
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-19: TRACE events summary

Code	Mnemonic	Name	Description
0x400C	TRB_WRAP	Trace buffer current write pointer wrapped	This event is generated each time the current write pointer is wrapped to the base pointer.
0x400E	TRB_TRIG	Trace buffer Trigger Event	This event is generated when a Trace Buffer Extension Trigger Event occurs.
0x4010	TRCEXTOUT0	Trace unit external output 0	This event is generated each time an event is signaled by ETE external event 0.
0x4011	TRCEXTOUT1	Trace unit external output 1	This event is generated each time an event is signaled by ETE external event 1.
0x4012	TRCEXTOUT2	Trace unit external output 2	This event is generated each time an event is signaled by ETE external event 2.
0x4013	TRCEXTOUT3	Trace unit external output 3	This event is generated each time an event is signaled by ETE external event 3.
0x4018	CTI_TRIGOUT4	Cross-trigger Interface output trigger 4	This event is generated each time an event is signaled on CTI output trigger 4.
0x4019	CTI_TRIGOUT5	Cross-trigger Interface output trigger 5	This event is generated each time an event is signaled on CTI output trigger 5.
0x401A	CTI_TRIGOUT6	Cross-trigger Interface output trigger 6	This event is generated each time an event is signaled on CTI output trigger 6.
0x401B	CTI_TRIGOUT7	Cross-trigger Interface output trigger 7	This event is generated each time an event is signaled on CTI output trigger 7.

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x400C TRB_WRAP, Trace buffer current write pointer wrapped, event

This event is generated each time the current write pointer is wrapped to the base pointer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[TRACE](#)

0x400E TRB_TRIG, Trace buffer Trigger Event, event

This event is generated when a Trace Buffer Extension Trigger Event occurs.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[TRACE](#)

0x4010 TRCEXTOUT0, Trace unit external output 0, event

This event is generated each time an event is signaled by ETE external event 0.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TRACE

0x4011 TRCEXTOUT1, Trace unit external output 1, event

This event is generated each time an event is signaled by ETE external event 1.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TRACE

0x4012 TRCEXTOUT2, Trace unit external output 2, event

This event is generated each time an event is signaled by ETE external event 2.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TRACE

0x4013 TRCEXTOUT3, Trace unit external output 3, event

This event is generated each time an event is signaled by ETE external event 3.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TRACE

0x4018 CTI_TRIGOUT4, Cross-trigger Interface output trigger 4, event

This event is generated each time an event is signaled on CTI output trigger 4.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TRACE

0x4019 CTI_TRIGOUT5, Cross-trigger Interface output trigger 5, event

This event is generated each time an event is signaled on CTI output trigger 5.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TRACE

0x401A CTI_TRIGOUT6, Cross-trigger Interface output trigger 6, event

This event is generated each time an event is signaled on CTI output trigger 6.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TRACE

0x401B CTI_TRIGOUT7, Cross-trigger Interface output trigger 7, event

This event is generated each time an event is signaled on CTI output trigger 7.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

TRACE

6.20 Events without a functional group for Neoverse N3

Events in this section do not have a functional group.

Table 6-20: No functional group events summary

Code	Mnemonic	Name	Description
0x015C	IMP_STALL_BACKEND_IQ_SX	Backend stall cycles, SX IQ full	"Counts each cycle counted by STALL_BACKEND_BUSY where the backend is stalled due to the SX IQ...
0x015D	IMP_STALL_BACKEND_IQ_MX	Backend stall cycles, MX IQ full	"Counts each cycle counted by STALL_BACKEND_BUSY where the backend is stalled due to the MX IQ...
0x015E	IMP_STALL_BACKEND_IQ_LS	Backend stall cycles, LS IQ full	"Counts each cycle counted by STALL_BACKEND_BUSY where the backend is stalled due to the LS IQ...
0x015F	IMP_STALL_BACKEND_IQ_VX	Backend stall cycles, VX IQ full	"Counts each cycle counted by STALL_BACKEND_BUSY where the backend is stalled due to the VX IQ...
0x0160	IMP_STALL_BACKEND_MCQ	Backend stall cycles, MCQ full	"Counts each cycle counted by STALL_BACKEND_CPUBOUND where the backend is stalled due to the MCQ...

Code	Mnemonic	Name	Description
0x0198	IMP_L2_CHI_CBUSY0	Number of RXDAT or RXRSP response received with CBusy of 0	Count of RXDAT or RXRSP responses received with indication completer fullness indicator set to 0
0x0199	IMP_L2_CHI_CBUSY1	Number of RXDAT or RXRSP response received with CBusy of 1	Count of RXDAT or RXRSP responses received with indication completer fullness indicator set to 1
0x019A	IMP_L2_CHI_CBUSY2	Number of RXDAT or RXRSP response received with CBusy of 2	Count of RXDAT or RXRSP responses received with indication completer fullness indicator set to 2
0x019B	IMP_L2_CHI_CBUSY3	Number of RXDAT or RXRSP response received with CBusy of 3	Count of RXDAT or RXRSP responses received with indication completer fullness indicator set to 3
0x019C	IMP_L2_CHI_CBUSY_MT	Number of RXDAT or RXRSP response received with CBusy Multi-threaded set	Count of RXDAT or RXRSP responses received with indication completer indicating multiple cores...
0x1003	IMP_STALL_BACKEND_PCRF	Backend stall cycles, PCRF full	"Counts each cycle counted by STALL_BACKEND_CPUBOUND where the backend is stalled due to the PCRF..."
0x1005	IMP_STALL_BACKEND_RSTACK	Backend stall cycles, return stack full	"Counts each cycle counted by STALL_BACKEND_CPUBOUND where the backend is stalled due to the..."

For a complete list of the events in Neoverse N3, see [PMU events cheat sheet for Neoverse N3](#) and [PMU events lookup table for Neoverse N3](#).

0x015c IMP_STALL_BACKEND_IQ_SX, Backend stall cycles, SX IQ full, event

"Counts each cycle counted by STALL_BACKEND_BUSY where the backend is stalled due to the SX IQ being full"

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x015d IMP_STALL_BACKEND_IQ_MX, Backend stall cycles, MX IQ full, event

"Counts each cycle counted by STALL_BACKEND_BUSY where the backend is stalled due to the MX IQ being full"

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x015e IMP_STALL_BACKEND_IQ_LS, Backend stall cycles, LS IQ full, event

"Counts each cycle counted by STALL_BACKEND_BUSY where the backend is stalled due to the LS IQ being full"

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x015F IMP_STALL_BACKEND_IQ_VX, Backend stall cycles, VX IQ full, event

“Counts each cycle counted by STALL_BACKEND_BUSY where the backend is stalled due to the VX IQ being full”

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x0160 IMP_STALL_BACKEND_MCQ, Backend stall cycles, MCQ full, event

“Counts each cycle counted by STALL_BACKEND_CPUBOUND where the backend is stalled due to the MCQ being full”

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x0198 IMP_L2_CHI_CBUSY0, Number of RXDAT or RXRSP response received with CBusy of 0, event

Count of RXDAT or RXRSP responses received with indication completer fullness indicator set to 0

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x0199 IMP_L2_CHI_CBUSY1, Number of RXDAT or RXRSP response received with CBusy of 1, event

Count of RXDAT or RXRSP responses received with indication completer fullness indicator set to 1

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x019A IMP_L2_CHI_CBUSY2, Number of RXDAT or RXRSP response received with CBusy of 2, event

Count of RXDAT or RXRSP responses received with indication completer fullness indicator set to 2

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x019B IMP_L2_CHI_CBUSY3, Number of RXDAT or RXRSP response received with CBusy of 3, event

Count of RXDAT or RXRSP responses received with indication completer fullness indicator set to 3

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x019C IMP_L2_CHI_CBUSY_MT, Number of RXDAT or RXRSP response received with CBusy Multi-threaded set, event

Count of RXDAT or RXRSP responses received with indication completer indicating multiple cores actively making requests

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x1003 IMP_STALL_BACKEND_PCRF, Backend stall cycles, PCRF full, event

“Counts each cycle counted by STALL_BACKEND_CPUBOUND where the backend is stalled due to the PCRF being full”

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

0x1005 IMP_STALL_BACKEND_RSTACK, Backend stall cycles, return stack full, event

“Counts each cycle counted by STALL_BACKEND_CPUBOUND where the backend is stalled due to the return stack being full”

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the Neoverse N3 Methodology Specification.

Functional groups

[Events without a functional group](#)

7. Supplemental performance debug PMU events

Neoverse N3 implements an additional set of events that are available for use in debugging the performance behaviors of the processor. These events are not guaranteed to have the same level of accuracy as the architected PMU counters. Any use of these events must take this variable accuracy into account.

The following table shows the events.

Table 7-1: Supplemental performance debug PMU events for Neoverse N3

Code	Mnemonic	Description
0x120	IMP_CT_FLUSH	Flushes including architectural, microarchitectural, and branch redirects
0x121	IMP_CT_FLUSH_MEM	Flushes due to memory hazards
0x122	IMP_CT_FLUSH_BAD_BRANCH	Flushes due to non-branch instruction predicted as a branch
0x124	IMP_CT_FLUSH_ISB	Flushes due to ISB or similar side-effects
0x125	IMP_CT_FLUSH_OTHER	Flushes due to other hazards
0x127	IMP_LS_RAR	Loadstore detected nuke due to read-after-read ordering hazard
0x128	IMP_LS_RAW	Loadstore detected nuke due to read-after-write ordering hazard
0x158	IMP_STALL_BACKEND_RENAME_FRF	RN dispatch stall due to flag registers
0x159	IMP_STALL_BACKEND_RENAME_GRF	RN dispatch stall due to general registers
0x15A	IMP_STALL_BACKEND_RENAME_VRF	RN dispatch stall due to vector registers

Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant

export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

The information in this document is Final, that is for a developed product.

Revision history

These sections can help you understand how the document has changed over time.

Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

Document history

Issue	Date	Confidentiality	Change
0100-01	23 September 2024	Non-Confidential	First issue for all revisions of the product.

Change history

The Change history tables describe the technical changes between released issues of this document in reverse order. Issue numbers match the revision history in [Document release information](#) on page 176.

Table 2: Issue 0100-01

Change	Location
First release for all revisions of the product.	-

Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
<i>italic</i>	Citations.
bold	Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: <div>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></div>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the Arm® Glossary. For example, IMPLEMENTATION DEFINED , IMPLEMENTATION SPECIFIC , UNKNOWN , and UNPREDICTABLE .



We recommend the following. If you do not follow these recommendations your system might not work.



Your system requires the following. If you do not follow these requirements your system will not work.



You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.



This information is important and needs your attention.



This information might help you perform a task in an easier, better, or faster way.



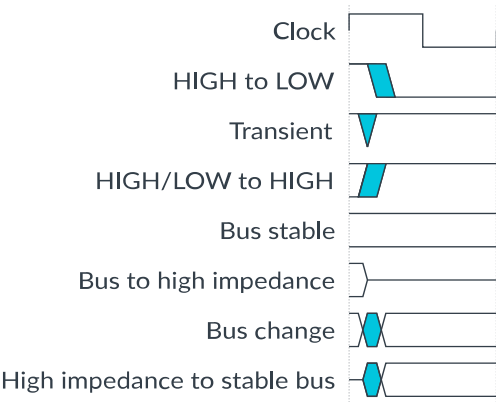
This information reminds you of something important relating to the current content.

Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

Figure 1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lowercase n

At the start or end of a signal name, n denotes an active-LOW signal.

Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm® CPU Telemetry Solution Topdown Methodology Specification	109542	Non-Confidential
Arm® Neoverse™ N3 Core Technical Reference Manual	107997	Non-Confidential
Arm® Telemetry Solution GitLab repository	–	Non-Confidential
Arm® Telemetry on Arm Developer	–	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Architecture Reference Manual for A-profile architecture	DDI 0487	Non-Confidential