



# Arm<sup>®</sup> Neoverse<sup>™</sup> N1 Core

Version 1.0

## Telemetry Specification

**Non-Confidential**

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

**Issue 01**

108070\_0100\_01\_en



Arm® Neoverse™ N1 Core  
Telemetry Specification

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

Release information

Document history

| Issue   | Date             | Confidentiality  | Change          |
|---------|------------------|------------------|-----------------|
| 0100-01 | 20 February 2024 | Non-Confidential | Initial release |

Proprietary Notice

This document is protected by copyright and other related rights and the practice 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. 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 implementations infringe any third party patents.

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, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

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.

This document consists solely of commercial items. You shall be responsible for ensuring that any 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 the Agreement shall prevail.

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. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>.

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

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

## Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

## Product Status

The information in this document is Final, that is for a developed product.

## 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>.

## 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](mailto:terms@arm.com).

# Contents

|   |           |
|---|-----------|
| <b>1. Introduction.....</b>   | <b>7</b>  |
| 1.1 Conventions.....  | 7         |
| 1.2 Useful resources.....   | 8         |
| 1.3 Other information.....  | 9         |
| <b>2. Overview of the Neoverse N1 Telemetry methodology.....</b>    | <b>10</b> |
| 2.1 Documentation and resources.....                                | 11        |
| <b>3. Telemetry features of Neoverse N1.....</b>                    | <b>12</b> |
| <b>4. CPU performance analysis methodology.....</b>                 | <b>14</b> |
| 4.1 Topdown methodology for Neoverse N1.....                        | 14        |
| 4.2 Stage 1: Topdown analysis.....                                  | 16        |
| 4.3 Stage 2: Microarchitecture exploration.....                     | 16        |
| <b>5. Neoverse N1 Telemetry cheat-sheets and lookup tables.....</b> | <b>21</b> |
| 5.1 Metrics cheat sheet for Neoverse N1.....                        | 21        |
| 5.2 PMU events cheat sheet for Neoverse N1.....                     | 22        |
| 5.3 Metrics lookup table for Neoverse N1.....                       | 24        |
| 5.4 PMU events lookup table for Neoverse N1.....                    | 26        |
| <b>6. Metrics by metric group in Neoverse N1.....</b>               | <b>33</b> |
| 6.1 Cycle_Accounting metrics for Neoverse N1.....                   | 33        |
| 6.2 General metrics for Neoverse N1.....                            | 35        |
| 6.3 MPKI metrics for Neoverse N1.....                               | 35        |
| 6.4 Miss_Ratio metrics for Neoverse N1.....                         | 40        |
| 6.5 Branch_Effectiveness metrics for Neoverse N1.....               | 46        |
| 6.6 ITLB_Effectiveness metrics for Neoverse N1.....                 | 47        |
| 6.7 DTLB_Effectiveness metrics for Neoverse N1.....                 | 51        |
| 6.8 L1I_Cache_Effectiveness metrics for Neoverse N1.....            | 54        |
| 6.9 L1D_Cache_Effectiveness metrics for Neoverse N1.....            | 56        |
| 6.10 L2_Cache_Effectiveness metrics for Neoverse N1.....            | 57        |
| 6.11 LL_Cache_Effectiveness metrics for Neoverse N1.....            | 58        |
| 6.12 Operation_Mix metrics for Neoverse N1.....                     | 60        |

**7. PMU events by functional group in Neoverse N1.....65**

7.1 Bus (BUS) events for Neoverse N1..... 65

7.2 Chain (CHAIN) events for Neoverse N1..... 67

7.3 Exception (EXCEPTION) events for Neoverse N1..... 68

7.4 L1D\_Cache (L1D CACHE) events for Neoverse N1.....72

7.5 L1I\_Cache (L1I CACHE) events for Neoverse N1.....76

7.6 L2\_Cache (L2 CACHE) events for Neoverse N1.....78

7.7 L3\_Cache (L3 CACHE) events for Neoverse N1.....82

7.8 LL\_Cache (LL CACHE) events for Neoverse N1.....83

7.9 Memory (MEMORY) events for Neoverse N1..... 85

7.10 Retired (RETIRED) events for Neoverse N1.....87

7.11 SPE (SPE) events for Neoverse N1..... 90

7.12 Spec\_Operation (SPEC OPERATION) events for Neoverse N1.....91

7.13 Stall (STALL) events for Neoverse N1..... 99

7.14 General (GENERAL) events for Neoverse N1..... 100

7.15 TLB (TLB) events for Neoverse N1..... 101

# 1. Introduction

## 1.1 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](https://developer.arm.com/glossary).

### Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

| Convention                 | Use  |
|----------------------------|--|
| <i>italic</i>              | Citations.   |
| <b>bold</b>                | Interface elements, such as menu names.<br><br>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.<br><br>For example:<br><div>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</div>      |
| SMALL CAPITALS             | Terms that have specific technical meanings as defined in the Arm® Glossary. For example, <b>IMPLEMENTATION DEFINED</b> , <b>IMPLEMENTATION SPECIFIC</b> , <b>UNKNOWN</b> , and <b>UNPREDICTABLE</b> . |



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 is important information and needs your attention.



A useful tip that might make it easier, better or faster to perform a task.



A reminder of something important that relates to the information you are reading.

## 1.2 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](https://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  |
|---|-------------|------------------|
| <a href="#">Arm® CPU Telemetry Solution Topdown Methodology Specification</a> | 109542      | Non-Confidential |
| <a href="#">Arm® Neoverse® N1 Core Technical Reference Manual</a>             | 100616      | Non-Confidential |
| <a href="#">Arm® Telemetry on Arm Developer</a>                               | –           | Non-Confidential |
| <a href="#">Arm® Telemetry Solution GitLab repository</a>                     | –           | Non-Confidential |



Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at <http://www.adobe.com>



## 1.3 Other information

See the Arm website for other relevant information.

- [Arm® Developer](#).
- [Arm® Documentation](#).
- [Technical Support](#).
- [Arm® Glossary](#).

## 2. Overview of the Neoverse N1 Telemetry methodology

The Arm® Neoverse N1 Core Telemetry Specification describes the Topdown methodology, derived metrics, and Performance Monitoring Unit (PMU) events supported by the Arm Neoverse N1 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 N1 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](#).

## 2.1 Documentation and resources

Arm products include a set of documents.

The documentation and resources for Neoverse N1 consist of:

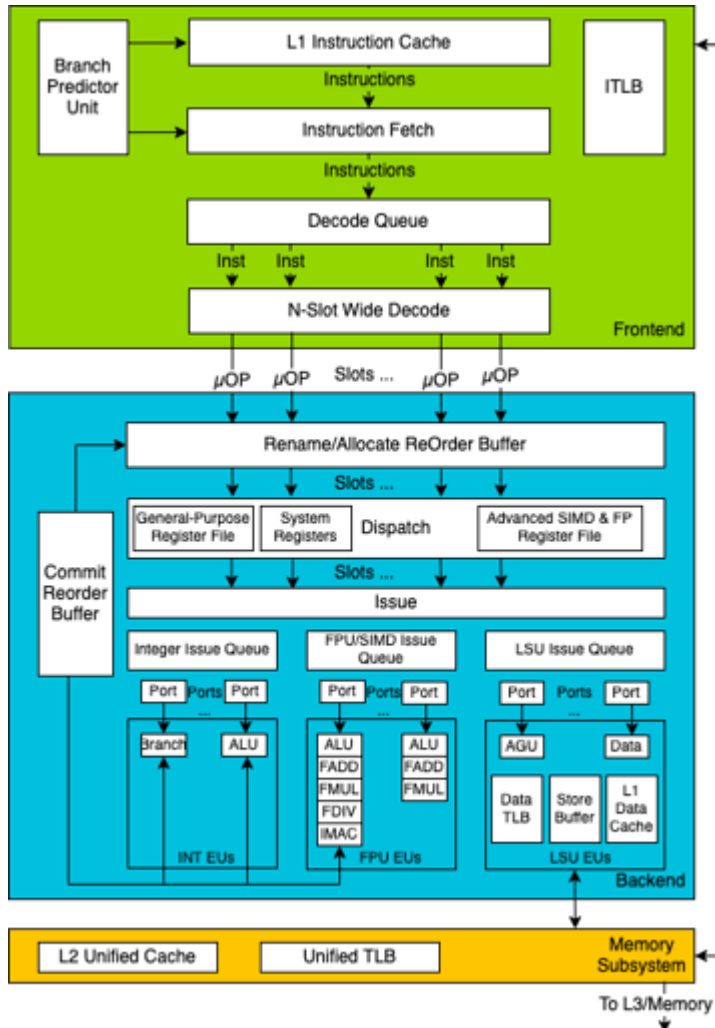
- [Arm® Telemetry on Arm Developer](#)
- [Arm® Telemetry Solution GitLab repository](#)
- [Arm® Neoverse® N1 Core Technical Reference Manual](#)

### 3. Telemetry features of Neoverse N1

The Neoverse N1 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 N1.

**Figure 3-1: Neoverse N1 core microarchitecture**



The frontend of the core is comprised of the instruction fetch and decode units. The frontend also includes a branch predictor unit that fetches instructions ahead of the pipeline and 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. The decode 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 bandwidth is determined by the number of renamed SLOTS available in the microarchitecture. 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 store 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 ( $\mu$ 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 Memory subsystem of the core handles the execution of load and store operations which rely heavily on the memory hierarchy levels. The Neoverse N1 has a dedicated L1/L2 cache 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 Neoverse N1 has multiple load/store units, which can both handle read and write operations. The core supports two hierarchical set associative caches, L1 Data Cache and L2 Cache whose size is configurable for each implementation. The private L2 cache of the core connects to the rest of the system through an AMBA® 5 CHI interface.

## Neoverse N1 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 N1

The Neoverse N1 PMU has six configurable counter registers and one dedicated function counter to count CPU cycles.

## 4. CPU performance analysis methodology

The Arm Topdown methodology for performance analysis and microarchitecture exploration is conducted in two stages.

### Stage 1

The first stage is to perform Topdown analysis. It uses hierarchical pipeline stall-related metrics to detect and identify the performance bottleneck in the CPU. For more information, see [Stage 1: Topdown analysis](#).

### Stage 2

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

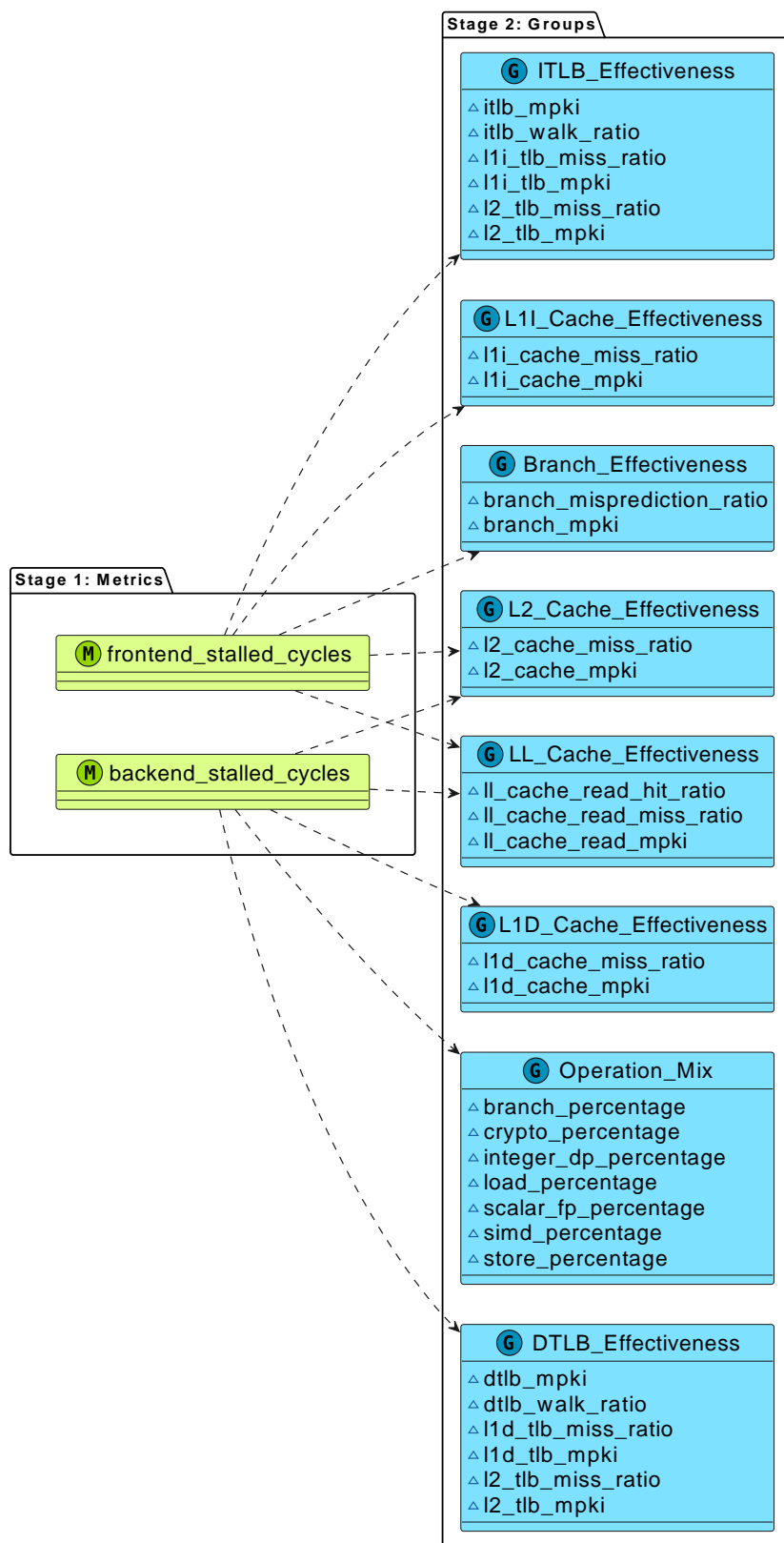
For more information, see [Arm® CPU Telemetry Solution Topdown Methodology Specification](#).

### 4.1 Topdown methodology for Neoverse N1

Topdown analysis helps with hotspot detection and is the first stage completed in the Topdown methodology. Microarchitecture exploration is the second stage completed in the Topdown methodology and conducts a micro-architectural analysis of the bottlenecking CPU resource.

The following figure shows the Topdown methodology for Neoverse N1 covering both the Stage 1 metrics and the Stage 2 metric groups that can be used during your analysis.

Figure 4-1: Neoverse N1 Topdown methodology overview



Arm recommends collecting all the metrics that are in Stage 1 and Stage 2 for workload characterization. For further analysis, Arm recommends a set of set of microarchitecture exploration metric groups against some of the hotspots detected in Stage 1. All the Stage 2 metrics can be used to derive further insights into the overall microarchitecture behavior during the execution of the application under investigation and can be used independently to Stage 1.

## 4.2 Stage 1: Topdown analysis

Neoverse N1 supports the two metrics for Topdown analysis level 1 that are cycle based.

The two metrics in the first level as part of the metric group CycleAccounting is defined by the [Arm® CPU Telemetry Solution Topdown Methodology Specification](#), as follows:

### **frontend\_stalled\_cycles**

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

### **backend\_stalled\_cycles**

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

For more information on the CycleAccounting metric group and its corresponding metrics, see [CycleAccounting](#).

## 4.3 Stage 2: Microarchitecture exploration

When the execution pipeline bottleneck region is identified from Stage 1, the next step is to investigate the CPU resources for further analysis.

As described in the Architecture Specification:

- A relatively high `frontend_stalled_cycles` metric shows that execution cycles are wasted due to pipeline stalls in the in-order frontend division of the processor. There are many reasons why frontend stalls can occur, such as inefficiency in the branch prediction unit, fetch latency due to instruction cache misses, and translation delays caused by Instruction TLB walks.
- A relatively high `backend_stalled_cycles` metric shows that execution cycles are wasted due to pipeline stalls in the backend of the processor. There are many reasons why backend stalls can occur, such as inefficiency in execution units, data cache misses, and translation delays caused by data TLB walks.

Neoverse N1 supports the following microarchitecture exploration metric groups that can be used for Stage 2 analysis. The metric groups here support further analysis of the bottlenecking CPU resources following the Stage 1 analysis hotspot.

As a common step in Stage 2, the two metric groups MPKI and Miss rate are recommended for a quick behavioral analysis of the CPU components that could be the most probable bottlenecks.



## MPKI – Misses Per Kilo Instructions

Misses Per Kilo Instructions is a set of metrics that can be derived to normalize the misses in CPU components, such as branches, caches, and TLBs against the total instructions executed. This is an industry-standard metric that also helps with comparison across different implementations of the Arm architecture, as instructions retired should count the same on all AArch64-based microarchitectures.

[MPKI](#) lists all the Misses Per Kilo Instructions metrics that can be derived for Neoverse N1 and their formulae.

## Miss Ratio

The Miss Ratio metric group provides a set of metrics that calculate the ratio of the misses in the CPU components, such as branches, caches and TLBs against the total accesses in those components. These metrics provide insights on the efficiency of each CPU component in the pipeline and help to root cause issues.

[Miss\\_Ratio](#) lists all the Miss Ratio metrics that can be derived for Neoverse N1 and their formulae.

## Operation Mix

The Neoverse N1 microarchitecture has a variety of execution units that process different types of operations.

The execution units that are shown in [Figure 3-1: Neoverse N1 core microarchitecture](#) on page 12 process the following types of operations:

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

The PMU events listed in [Spec\\_Operation](#) count the operations that are issued to these execution units.

[Operation\\_Mix](#) lists all the Speculative Operation Mix metrics that can be derived for Neoverse N1 and their formulae.

These metrics use events that count speculatively issued operations at the issue stage, which provide an estimate of the execution unit utilization, but not the retired instruction mix of the program. To derive the utilization of each operation type, the percentage of each type of operation is calculated as a fraction of the total operations issued, which is counted by the event INST\_SPEC.

Neoverse N1 does not support retired events for counting the architectural instruction mix. Neoverse N1 supports events to further break down the branch operations into immediate, indirect, and return branches, counted by events BR\_IMMED\_SPEC, BR\_INDIRECT\_SPEC, and BR\_RETURN\_SPEC respectively. BR\_RETURN\_SPEC is a subset of BR\_INDIRECT\_SPEC, as returns

are also counted as indirect branches. The sum of the BR\_IMMED\_SPEC and BR\_INDIRECT\_SPEC branch operation events can compute the total branches executed.

## Branch Effectiveness and Branch Mix

Branch mispredictions are costly in a deeply pipelined CPU, causing pipeline flushes and wasted cycles. As a general rule, workloads typically contain, on average, one branch in every six instructions.

Though modern CPUs have optimized branch prediction units, there are many use cases such as ray tracing and decision tree algorithms that are branch heavy and hard to predict. In some of these applications, there can be hundreds of unique branch paths to take and the target may be input data dependent.

Branch prediction performance can be evaluated using two PMU events, BR\_MIS\_PRED\_RETIRED and BR\_RETIRED. BR\_MIS\_PRED\_RETIRED provides an account of the total branches that were executed but mispredicted. This means that the direction of the code path was wrong and the following operations in the path were wasted, causing a pipeline flush. BR\_RETIRED counts the total branches architecturally executed by the CPU.

There are two performance metrics that can be derived for a high-level evaluation of the branch execution performance regarding the overall program execution:

### branch\_mпки metrics

Provides total branch mispredictions per kilo instructions

### branch\_misprediction\_ratio metrics

Provides an indication of the ratio of branches that were mispredicted to overall branches

[Branch\\_Effectiveness](#) lists all the Branch Effectiveness metrics that can be derived for Neoverse N1 and their formulae.

Branch prediction units work differently depending on the branch type. The following list describes the three main sub-units that work for different branch types:

- *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.

Neoverse N1 supports the following three events that respectively categorizes the immediate, indirect, and return branches executed:

- BR\_IMMED\_SPEC
- BR\_RETURN\_SPEC
- BR\_INDIRECT\_SPEC

Getting a breakdown of the branch type helps to investigate each of these sub-units within the branch prediction unit.

## TLB/MMU Effectiveness

Arm recommends checking the virtual memory system performance which affects the instruction fetch performance in the frontend and memory access performance on the data side.

The processor needs to translate a virtual address to physical address for any instruction/data memory access before it accesses the respective cache. A program's view of memory is virtual address, but the processor works with the physical address when accessing cache or memory.

Virtual to physical mappings are defined in the page translation tables which reside in system memory. Accessing these tables requires one or more memory accesses that take many cycles to complete, which is referred to as a translation table walk. However, to make these translations faster, *Translation Lookaside Buffers* (TLBs) cache translation table walks, greatly reducing the number of accesses to system memory.

Neoverse N1 implements a two-level TLB hierarchy. The first level contains separate, dedicated TLBs for the instruction and data (load/store) address translations. Total accesses to these TLBs are counted by L1I\_TLB and L1D\_TLB respectively. The second level contains a unified L2 TLB that is shared by both instruction-side and data-side accesses. There are corresponding REFILL counters for these TLB levels. Some performance metrics that can be derived for a high-level evaluation of the TLB execution performance are the `l<n>_tlb_mpki` and `l<n>_tlb_miss_rate` metrics, where `<n>` stands for each level of TLB instruction and data side.

Accesses that cause a translation table walk due to misses in the instruction side and data side TLBs are counted by the events `ITLB_WALK` and `DTLB_WALK`, respectively. To evaluate the TLB effectiveness and cost of latency caused by translation table walks specifically, `dtlb_mpki`, `dtlb_walk_ratio`, `itlb_mpki`, and `itlb_walk_ratio` are the key metrics that can be derived. `itlb_mpki` and `dtlb_mpki` provide the rate of TLB Walks per kilo instructions for instruction and data accesses respectively. These derived metrics help to evaluate and correlate the TLB efficiency with respect to the total instructions.

`dtlb_walk_ratio` provides the ratio of DTLB Walks to the overall TLB lookups made by the program. This is the same as `DTLB_WALK/MEM_ACCESS` as every `MEM_ACCESS` causes a `L1D_TLB` access. `itlb_walk_ratio` provides a percentage of ITLB walks to the overall TLB lookups initiated from the instruction side.

[ITLB\\_Effectiveness](#) and [DTLB\\_Effectiveness](#) list all the TLB Effectiveness metrics that can be derived for Neoverse N1 and their formulae.

## Cache Effectiveness

The Neoverse N1 implements a multi-level cache hierarchy.

The first level (L1) includes a dedicated cache for instructions and a separate dedicated cache for data accesses. The second level (L2) is a unified L2 cache that is shared between code and data. Further down the hierarchy, the system could have an optional shared *System Level Cache* (SLC) in the interconnect. Arm recommends checking the cache configurations with the platform provider.

The Neoverse N1 core supports hierarchical PMU events for all the cache hierarchy levels. For each level of caches, there are total access counts and refill counts. AArch64 does not support

cache MISS counters, but only REFILLS. A cache miss could lead to multiple cache line refills if the access is on a cache line boundary or multiple cache misses could be satisfied by a single REFILL.

Some performance metrics that can be derived for a high-level evaluation of the cache execution behavior are the  $I_{<n>\_cache\_mpki}$  and  $I_{<n>\_cache\_miss\_ratio}$  metrics, where  $<n>$  stands for each level of instruction and data caches.

[L1I\\_Cache\\_Effectiveness](#), [L1D\\_Cache\\_Effectiveness](#), and [L2\\_Cache\\_Effectiveness](#) list all the L1 and L2 Cache Effectiveness metrics that can be derived for Neoverse N1 and their formulae.

## 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. As these operations are looked up in the L1D\_CACHE first, both the events L1D\_CACHE and MEM\_ACCESS count at the same rate.

Neoverse N1 also supports two additional events, MEM\_ACCESS\_RD and MEM\_ACCESS\_WR that can provide the read and write traffic breakdown respectively. Note that these events are not the same as LD\_SPEC and ST\_SPEC as they count memory operations speculatively issued but not necessarily executed.

## Last Level Cache Counter Usage

On systems that support a shared SLC in the interconnect, LL\_CACHE\_RD counts the total accesses to the SLC. In a system that has the SLC configured to count LL\_CACHE\_RD events, LL\_CACHE\_RD counts total SLC accesses made by the core, and LL\_CACHE\_MISS\_RD counts accesses missed at the SLC.

The last level cache read miss metrics  $II\_cache\_read\_mpki$  and  $II\_cache\_miss\_ratio$  can be derived to study the last level read behavior. Another useful metric to measure is the SLC hit percentage for the read traffic is the SLC Read Hit Ratio denoted as  $II\_cache\_read\_hit\_ratio$ . Last level cache events do not have a write variant in Neoverse N1, since 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 not necessarily completed.

[LL\\_Cache\\_Effectiveness](#) lists all the Last Level Cache Effectiveness metrics that can be derived for Neoverse N1 and their formulae.

## Remote Cache Access

For systems with multiple sockets or SoCs, Neoverse N1 supports the REMOTE\_ACCESS event, which counts the memory transactions that were completed by a subordinate source from another chip.

## 5. Neoverse N1 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.

### 5.1 Metrics cheat sheet for Neoverse N1

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

Neoverse N1 specification provides the following types of metrics:

- Total implemented Common metrics: 31

| Cycle Accounting (2)   | General (1)   | Misses Per Kilo Instructions (10)  |
|--|---|--|
| <ul style="list-style-type: none"><li>• <a href="#">backend_stalled_cycles</a></li><li>• <a href="#">frontend_stalled_cycles</a></li></ul> | <ul style="list-style-type: none"><li>• <a href="#">ipc</a></li></ul> | <ul style="list-style-type: none"><li>• <a href="#">branch_mpki</a></li><li>• <a href="#">dtlb_mpki</a></li><li>• <a href="#">itlb_mpki</a></li><li>• <a href="#">l1d_cache_mpki</a></li><li>• <a href="#">l1d_tlb_mpki</a></li><li>• <a href="#">l1i_cache_mpki</a></li><li>• <a href="#">l1i_tlb_mpki</a></li><li>• <a href="#">l2_cache_mpki</a></li><li>• <a href="#">l2_tlb_mpki</a></li><li>• <a href="#">ll_cache_read_mpki</a></li></ul> |

| Miss Ratio (10)  | Branch Effectiveness (2)  | Instruction TLB Effectiveness (6)   |
|--|---|---|
| <ul style="list-style-type: none"> <li>branch_misprediction_ratio</li> <li>dtlb_walk_ratio</li> <li>itlb_walk_ratio</li> <li>l1d_cache_miss_ratio</li> <li>l1d_tlb_miss_ratio</li> <li>l1i_cache_miss_ratio</li> <li>l1i_tlb_miss_ratio</li> <li>l2_cache_miss_ratio</li> <li>l2_tlb_miss_ratio</li> <li>ll_cache_read_miss_ratio</li> </ul> | <ul style="list-style-type: none"> <li>branch_misprediction_ratio</li> <li>branch_mpki</li> </ul>                                       | <ul style="list-style-type: none"> <li>itlb_mpki</li> <li>itlb_walk_ratio</li> <li>l1i_tlb_miss_ratio</li> <li>l1i_tlb_mpki</li> <li>l2_tlb_miss_ratio</li> <li>l2_tlb_mpki</li> </ul>  |
| Data TLB Effectiveness (6)   | L1 Instruction Cache Effectiveness (2)  | L1 Data Cache Effectiveness (2)   |
| <ul style="list-style-type: none"> <li>dtlb_mpki</li> <li>dtlb_walk_ratio</li> <li>l1d_tlb_miss_ratio</li> <li>l1d_tlb_mpki</li> <li>l2_tlb_miss_ratio</li> <li>l2_tlb_mpki</li> </ul>   | <ul style="list-style-type: none"> <li>l1i_cache_miss_ratio</li> <li>l1i_cache_mpki</li> </ul>  | <ul style="list-style-type: none"> <li>l1d_cache_miss_ratio</li> <li>l1d_cache_mpki</li> </ul>  |
| L2 Unified Cache Effectiveness (2)   | Last Level Cache Effectiveness (3)  | Speculative Operation Mix (7)   |
| <ul style="list-style-type: none"> <li>l2_cache_miss_ratio</li> <li>l2_cache_mpki</li> </ul>   | <ul style="list-style-type: none"> <li>ll_cache_read_hit_ratio</li> <li>ll_cache_read_miss_ratio</li> <li>ll_cache_read_mpki</li> </ul> | <ul style="list-style-type: none"> <li>branch_percentage</li> <li>crypto_percentage</li> <li>integer_dp_percentage</li> <li>load_percentage</li> <li>scalar_fp_percentage</li> <li>simd_percentage</li> <li>store_percentage</li> </ul> |

## 5.2 PMU events cheat sheet for Neoverse N1

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 N1 specification provides the following types of PMU events:

- Total implemented Common events: 110
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

| Cycle Accounting (3)   | General (2)   | Misses Per Kilo Instructions (11)   |
|--|---|---|
| <ul style="list-style-type: none"> <li>CPU_CYCLES</li> <li>STALL_BACKEND</li> <li>STALL_FRONTEND</li> </ul>  | <ul style="list-style-type: none"> <li>CPU_CYCLES</li> <li>INST_RETIRED</li> </ul>                              | <ul style="list-style-type: none"> <li>BR_MIS_PRED_RETIRED</li> <li>DTLB_WALK</li> <li>INST_RETIRED</li> <li>ITLB_WALK</li> <li>L1D_CACHE_REFILL</li> <li>L1D_TLB_REFILL</li> <li>L1I_CACHE_REFILL</li> <li>L1I_TLB_REFILL</li> <li>L2D_CACHE_REFILL</li> <li>L2D_TLB_REFILL</li> <li>LL_CACHE_MISS_RD</li> </ul> |
| Miss Ratio (18)  | Branch Effectiveness (3)  | Instruction TLB Effectiveness (6)   |
| <ul style="list-style-type: none"> <li>BR_MIS_PRED_RETIRED</li> <li>BR_RETIRED</li> <li>DTLB_WALK</li> <li>ITLB_WALK</li> <li>L1D_CACHE</li> <li>L1D_CACHE_REFILL</li> <li>L1D_TLB</li> <li>L1D_TLB_REFILL</li> <li>L1I_CACHE</li> <li>L1I_CACHE_REFILL</li> <li>L1I_TLB</li> <li>L1I_TLB_REFILL</li> <li>L2D_CACHE</li> <li>L2D_CACHE_REFILL</li> <li>L2D_TLB</li> <li>L2D_TLB_REFILL</li> <li>LL_CACHE_MISS_RD</li> <li>LL_CACHE_RD</li> </ul> | <ul style="list-style-type: none"> <li>BR_MIS_PRED_RETIRED</li> <li>BR_RETIRED</li> <li>INST_RETIRED</li> </ul> | <ul style="list-style-type: none"> <li>INST_RETIRED</li> <li>ITLB_WALK</li> <li>L1I_TLB</li> <li>L1I_TLB_REFILL</li> <li>L2D_TLB</li> <li>L2D_TLB_REFILL</li> </ul>   |
| Data TLB Effectiveness (6)   | L1 Instruction Cache Effectiveness (3)  | L1 Data Cache Effectiveness (3)   |
| <ul style="list-style-type: none"> <li>DTLB_WALK</li> <li>INST_RETIRED</li> <li>L1D_TLB</li> <li>L1D_TLB_REFILL</li> <li>L2D_TLB</li> <li>L2D_TLB_REFILL</li> </ul>  | <ul style="list-style-type: none"> <li>INST_RETIRED</li> <li>L1I_CACHE</li> <li>L1I_CACHE_REFILL</li> </ul>     | <ul style="list-style-type: none"> <li>INST_RETIRED</li> <li>L1D_CACHE</li> <li>L1D_CACHE_REFILL</li> </ul>   |

| L2 Unified Cache Effectiveness (3)  | Last Level Cache Effectiveness (3)  | Speculative Operation Mix (9)  |
|---|---|--|
| <ul style="list-style-type: none"> <li>INST_RETIRED</li> <li>L2D_CACHE</li> <li>L2D_CACHE_REFILL</li> </ul> | <ul style="list-style-type: none"> <li>INST_RETIRED</li> <li>LL_CACHE_MISS_RD</li> <li>LL_CACHE_RD</li> </ul> | <ul style="list-style-type: none"> <li>ASE_SPEC</li> <li>BR_IMMED_SPEC</li> <li>BR_INDIRECT_SPEC</li> <li>CRYPTO_SPEC</li> <li>DP_SPEC</li> <li>INST_SPEC</li> <li>LD_SPEC</li> <li>ST_SPEC</li> <li>VFP_SPEC</li> </ul> |

## 5.3 Metrics lookup table for Neoverse N1

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 5-9: Metrics listed by name, with related events and metric groups**

| Metric Name  | Formula from Events                                  | Metric Groups  |
|--|--|--|
| backend_stalled_cycles   | STALL_BACKEND / CPU_CYCLES * 100                     | <ul style="list-style-type: none"> <li>Cycle_Accounting</li> </ul>                         |
| <ul style="list-style-type: none"> <li>branch_misprediction_ratio in Branch_Effectiveness</li> <li>branch_misprediction_ratio in Miss_Ratio</li> </ul> | BR_MIS_PRED_RETIRED / BR_RETIRED                     | <ul style="list-style-type: none"> <li>Branch_Effectiveness</li> <li>Miss_Ratio</li> </ul> |
| <ul style="list-style-type: none"> <li>branch_mpki in Branch_Effectiveness</li> <li>branch_mpki in MPKI</li> </ul>                                     | BR_MIS_PRED_RETIRED / INST_RETIRED * 1000            | <ul style="list-style-type: none"> <li>Branch_Effectiveness</li> <li>MPKI</li> </ul>       |
| branch_percentage  | (BR_IMMED_SPEC + BR_INDIRECT_SPEC) / INST_SPEC * 100 | <ul style="list-style-type: none"> <li>Operation_Mix</li> </ul>                            |
| crypto_percentage  | CRYPTO_SPEC / INST_SPEC * 100                        | <ul style="list-style-type: none"> <li>Operation_Mix</li> </ul>                            |
| <ul style="list-style-type: none"> <li>dtlb_mpki in DTLB_Effectiveness</li> <li>dtlb_mpki in MPKI</li> </ul>   | DTLB_WALK / INST_RETIRED * 1000                      | <ul style="list-style-type: none"> <li>DTLB_Effectiveness</li> <li>MPKI</li> </ul>         |
| <ul style="list-style-type: none"> <li>dtlb_walk_ratio in DTLB_Effectiveness</li> <li>dtlb_walk_ratio in Miss_Ratio</li> </ul>                         | DTLB_WALK / L1D_TLB                                  | <ul style="list-style-type: none"> <li>DTLB_Effectiveness</li> <li>Miss_Ratio</li> </ul>   |
| frontend_stalled_cycles  | STALL_FRONTEND / CPU_CYCLES * 100                    | <ul style="list-style-type: none"> <li>Cycle_Accounting</li> </ul>                         |
| integer_dp_percentage  | DP_SPEC / INST_SPEC * 100                            | <ul style="list-style-type: none"> <li>Operation_Mix</li> </ul>                            |
| ipc  | INST_RETIRED / CPU_CYCLES                            | <ul style="list-style-type: none"> <li>General</li> </ul>                                  |
| <ul style="list-style-type: none"> <li>itlb_mpki in ITLB_Effectiveness</li> <li>itlb_mpki in MPKI</li> </ul>   | ITLB_WALK / INST_RETIRED * 1000                      | <ul style="list-style-type: none"> <li>ITLB_Effectiveness</li> <li>MPKI</li> </ul>         |
| <ul style="list-style-type: none"> <li>itlb_walk_ratio in ITLB_Effectiveness</li> <li>itlb_walk_ratio in Miss_Ratio</li> </ul>                         | ITLB_WALK / L1I_TLB                                  | <ul style="list-style-type: none"> <li>ITLB_Effectiveness</li> <li>Miss_Ratio</li> </ul>   |



| Metric Name   | Formula from Events                                     | Metric Groups  |
|---|---|--|
| <ul style="list-style-type: none"> <li>l1d_cache_miss_ratio in L1D_Cache_Effectiveness</li> <li>l1d_cache_miss_ratio in Miss_Ratio</li> </ul>                                       | $L1D\_CACHE\_REFILL / L1D\_CACHE$                       | <ul style="list-style-type: none"> <li>L1D_Cache_Effectiveness</li> <li>Miss_Ratio</li> </ul>                        |
| <ul style="list-style-type: none"> <li>l1d_cache_mpki in L1D_Cache_Effectiveness</li> <li>l1d_cache_mpki in MPKI</li> </ul>   | $L1D\_CACHE\_REFILL / INST\_RETIRED * 1000$             | <ul style="list-style-type: none"> <li>L1D_Cache_Effectiveness</li> <li>MPKI</li> </ul>                              |
| <ul style="list-style-type: none"> <li>l1d_tlb_miss_ratio in DTLB_Effectiveness</li> <li>l1d_tlb_miss_ratio in Miss_Ratio</li> </ul>  | $L1D\_TLB\_REFILL / L1D\_TLB$                           | <ul style="list-style-type: none"> <li>DTLB_Effectiveness</li> <li>Miss_Ratio</li> </ul>                             |
| <ul style="list-style-type: none"> <li>l1d_tlb_mpki in DTLB_Effectiveness</li> <li>l1d_tlb_mpki in MPKI</li> </ul>  | $L1D\_TLB\_REFILL / INST\_RETIRED * 1000$               | <ul style="list-style-type: none"> <li>DTLB_Effectiveness</li> <li>MPKI</li> </ul>                                   |
| <ul style="list-style-type: none"> <li>l1i_cache_miss_ratio in L1I_Cache_Effectiveness</li> <li>l1i_cache_miss_ratio in Miss_Ratio</li> </ul>                                       | $L1I\_CACHE\_REFILL / L1I\_CACHE$                       | <ul style="list-style-type: none"> <li>L1I_Cache_Effectiveness</li> <li>Miss_Ratio</li> </ul>                        |
| <ul style="list-style-type: none"> <li>l1i_cache_mpki in L1I_Cache_Effectiveness</li> <li>l1i_cache_mpki in MPKI</li> </ul>   | $L1I\_CACHE\_REFILL / INST\_RETIRED * 1000$             | <ul style="list-style-type: none"> <li>L1I_Cache_Effectiveness</li> <li>MPKI</li> </ul>                              |
| <ul style="list-style-type: none"> <li>l1i_tlb_miss_ratio in ITLB_Effectiveness</li> <li>l1i_tlb_miss_ratio in Miss_Ratio</li> </ul>  | $L1I\_TLB\_REFILL / L1I\_TLB$                           | <ul style="list-style-type: none"> <li>ITLB_Effectiveness</li> <li>Miss_Ratio</li> </ul>                             |
| <ul style="list-style-type: none"> <li>l1i_tlb_mpki in ITLB_Effectiveness</li> <li>l1i_tlb_mpki in MPKI</li> </ul>  | $L1I\_TLB\_REFILL / INST\_RETIRED * 1000$               | <ul style="list-style-type: none"> <li>ITLB_Effectiveness</li> <li>MPKI</li> </ul>                                   |
| <ul style="list-style-type: none"> <li>l2_cache_miss_ratio in L2_Cache_Effectiveness</li> <li>l2_cache_miss_ratio in Miss_Ratio</li> </ul>  | $L2D\_CACHE\_REFILL / L2D\_CACHE$                       | <ul style="list-style-type: none"> <li>L2_Cache_Effectiveness</li> <li>Miss_Ratio</li> </ul>                         |
| <ul style="list-style-type: none"> <li>l2_cache_mpki in L2_Cache_Effectiveness</li> <li>l2_cache_mpki in MPKI</li> </ul>  | $L2D\_CACHE\_REFILL / INST\_RETIRED * 1000$             | <ul style="list-style-type: none"> <li>L2_Cache_Effectiveness</li> <li>MPKI</li> </ul>                               |
| <ul style="list-style-type: none"> <li>l2_tlb_miss_ratio in ITLB_Effectiveness</li> <li>l2_tlb_miss_ratio in DTLB_Effectiveness</li> <li>l2_tlb_miss_ratio in Miss_Ratio</li> </ul> | $L2D\_TLB\_REFILL / L2D\_TLB$                           | <ul style="list-style-type: none"> <li>ITLB_Effectiveness</li> <li>DTLB_Effectiveness</li> <li>Miss_Ratio</li> </ul> |
| <ul style="list-style-type: none"> <li>l2_tlb_mpki in ITLB_Effectiveness</li> <li>l2_tlb_mpki in DTLB_Effectiveness</li> <li>l2_tlb_mpki in MPKI</li> </ul>                         | $L2D\_TLB\_REFILL / INST\_RETIRED * 1000$               | <ul style="list-style-type: none"> <li>ITLB_Effectiveness</li> <li>DTLB_Effectiveness</li> <li>MPKI</li> </ul>       |
| ll_cache_read_hit_ratio   | $(LL\_CACHE\_RD - LL\_CACHE\_MISS\_RD) / LL\_CACHE\_RD$ | <ul style="list-style-type: none"> <li>LL_Cache_Effectiveness</li> </ul>   |
| <ul style="list-style-type: none"> <li>ll_cache_read_miss_ratio in LL_Cache_Effectiveness</li> <li>ll_cache_read_miss_ratio in Miss_Ratio</li> </ul>                                | $LL\_CACHE\_MISS\_RD / LL\_CACHE\_RD$                   | <ul style="list-style-type: none"> <li>LL_Cache_Effectiveness</li> <li>Miss_Ratio</li> </ul>                         |
| <ul style="list-style-type: none"> <li>ll_cache_read_mpki in LL_Cache_Effectiveness</li> <li>ll_cache_read_mpki in MPKI</li> </ul>  | $LL\_CACHE\_MISS\_RD / INST\_RETIRED * 1000$            | <ul style="list-style-type: none"> <li>LL_Cache_Effectiveness</li> <li>MPKI</li> </ul>                               |
| load_percentage   | $LD\_SPEC / INST\_SPEC * 100$                           | <ul style="list-style-type: none"> <li>Operation_Mix</li> </ul>  |
| scalar_fp_percentage  | $VFP\_SPEC / INST\_SPEC * 100$                          | <ul style="list-style-type: none"> <li>Operation_Mix</li> </ul>  |
| simd_percentage   | $ASE\_SPEC / INST\_SPEC * 100$                          | <ul style="list-style-type: none"> <li>Operation_Mix</li> </ul>  |

| Metric Name      | Formula from Events       | Metric Groups   |
|------------------|---------------------------|---|
| store_percentage | ST_SPEC / INST_SPEC * 100 | <ul style="list-style-type: none"> <li>Operation_Mix</li> </ul> |

## 5.4 PMU events lookup table for Neoverse N1

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: 754
- Total implemented Common events: 110
  - Common : Architectural-defined events: 46
  - Common : Implementation-defined events: 64
- Total Implemented Product ImpDef events: 0
- PMU Only Events : 0
- ETE Only Events : 0

**Table 5-10: 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"> <li>Retired</li> </ul>   |
| 0x0001, L1I_CACHE_REFILL | <ul style="list-style-type: none"> <li>l1i_cache_mpki in L1I_Cache_Effectiveness</li> <li>l1i_cache_mpki in MPKI</li> <li>l1i_cache_miss_ratio in L1I_Cache_Effectiveness</li> <li>l1i_cache_miss_ratio in Miss_Ratio</li> </ul> | <ul style="list-style-type: none"> <li>L1I_Cache_Effectiveness</li> <li>MPKI</li> <li>Miss_Ratio</li> </ul> | <ul style="list-style-type: none"> <li>L1I_Cache</li> </ul> |
| 0x0002, L1I_TLB_REFILL   | <ul style="list-style-type: none"> <li>l1i_tlb_mpki in ITLB_Effectiveness</li> <li>l1i_tlb_mpki in MPKI</li> <li>l1i_tlb_miss_ratio in ITLB_Effectiveness</li> <li>l1i_tlb_miss_ratio in Miss_Ratio</li> </ul>                   | <ul style="list-style-type: none"> <li>Miss_Ratio</li> <li>MPKI</li> <li>ITLB_Effectiveness</li> </ul>      | <ul style="list-style-type: none"> <li>TLB</li> </ul>       |
| 0x0003, L1D_CACHE_REFILL | <ul style="list-style-type: none"> <li>l1d_cache_mpki in L1D_Cache_Effectiveness</li> <li>l1d_cache_mpki in MPKI</li> <li>l1d_cache_miss_ratio in L1D_Cache_Effectiveness</li> <li>l1d_cache_miss_ratio in Miss_Ratio</li> </ul> | <ul style="list-style-type: none"> <li>L1D_Cache_Effectiveness</li> <li>MPKI</li> <li>Miss_Ratio</li> </ul> | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |

| Code, Mnemonic         | Metrics  | Metric Groups   | Functional Groups   |
|------------------------|--|---|---|
| 0x0004, L1D_CACHE      | <ul style="list-style-type: none"> <li>l1d_cache_miss_ratio in L1D_Cache_Effectiveness</li> <li>l1d_cache_miss_ratio in Miss_Ratio</li> </ul>  | <ul style="list-style-type: none"> <li>L1D_Cache_Effectiveness</li> <li>Miss_Ratio</li> </ul>   | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x0005, L1D_TLB_REFILL | <ul style="list-style-type: none"> <li>l1d_tlb_mpki in DTLB_Effectiveness</li> <li>l1d_tlb_mpki in MPKI</li> <li>l1d_tlb_miss_ratio in DTLB_Effectiveness</li> <li>l1d_tlb_miss_ratio in Miss_Ratio</li> </ul>   | <ul style="list-style-type: none"> <li>DTLB_Effectiveness</li> <li>MPKI</li> <li>Miss_Ratio</li> </ul>  | <ul style="list-style-type: none"> <li>TLB</li> </ul>       |
| 0x0008, INST_RETIRED   | <ul style="list-style-type: none"> <li>ipc</li> <li>branch_mpki in Branch_Effectiveness</li> <li>branch_mpki in MPKI</li> <li>itlb_mpki in ITLB_Effectiveness</li> <li>itlb_mpki in MPKI</li> <li>l1i_tlb_mpki in ITLB_Effectiveness</li> <li>l1i_tlb_mpki in MPKI</li> <li>dtlb_mpki in DTLB_Effectiveness</li> <li>dtlb_mpki in MPKI</li> <li>l1d_tlb_mpki in DTLB_Effectiveness</li> <li>l1d_tlb_mpki in MPKI</li> <li>l2_tlb_mpki in ITLB_Effectiveness</li> <li>l2_tlb_mpki in DTLB_Effectiveness</li> <li>l2_tlb_mpki in MPKI</li> <li>l1i_cache_mpki in L1I_Cache_Effectiveness</li> <li>l1i_cache_mpki in MPKI</li> <li>l1d_cache_mpki in L1D_Cache_Effectiveness</li> <li>l1d_cache_mpki in MPKI</li> <li>l2_cache_mpki in L2_Cache_Effectiveness</li> <li>l2_cache_mpki in MPKI</li> <li>ll_cache_read_mpki in LL_Cache_Effectiveness</li> <li>ll_cache_read_mpki in MPKI</li> </ul> | <ul style="list-style-type: none"> <li>LL_Cache_Effectiveness</li> <li>MPKI</li> <li>L1D_Cache_Effectiveness</li> <li>L2_Cache_Effectiveness</li> <li>Branch_Effectiveness</li> <li>General</li> <li>ITLB_Effectiveness</li> <li>L1I_Cache_Effectiveness</li> <li>DTLB_Effectiveness</li> </ul> | <ul style="list-style-type: none"> <li>Retired</li> </ul>   |
| 0x0009, EXC_TAKEN      | -  | -   | <ul style="list-style-type: none"> <li>Exception</li> </ul> |
| 0x000A, EXC_RETURN     | -  | -   | <ul style="list-style-type: none"> <li>Exception</li> </ul> |

| Code, Mnemonic             | Metrics   | Metric Groups  | Functional Groups  |
|----------------------------|---|--|--|
| 0x000B, CID_WRITE_RETIRED  | -   | -  | <ul style="list-style-type: none"> <li>Retired</li> </ul>        |
| 0x0010, BR_MIS_PRED        | -   | -  | <ul style="list-style-type: none"> <li>Spec_Operation</li> </ul> |
| 0x0011, CPU_CYCLES         | <ul style="list-style-type: none"> <li>frontend_stalled_cycles</li> <li>backend_stalled_cycles</li> <li>ipc</li> </ul>  | <ul style="list-style-type: none"> <li>Cycle_Accounting</li> <li>General</li> </ul>                        | <ul style="list-style-type: none"> <li>General</li> </ul>        |
| 0x0012, BR_PRED            | -   | -  | <ul style="list-style-type: none"> <li>Spec_Operation</li> </ul> |
| 0x0013, MEM_ACCESS         | -   | -  | <ul style="list-style-type: none"> <li>Memory</li> </ul>         |
| 0x0014, L1I_CACHE          | <ul style="list-style-type: none"> <li>l1i_cache_miss_ratio in L1_Cache_Effectiveness</li> <li>l1i_cache_miss_ratio in Miss_Ratio</li> </ul>  | <ul style="list-style-type: none"> <li>L1I_Cache_Effectiveness</li> <li>Miss_Ratio</li> </ul>              | <ul style="list-style-type: none"> <li>L1I_Cache</li> </ul>      |
| 0x0015, L1D_CACHE_WB       | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul>      |
| 0x0016, L2D_CACHE          | <ul style="list-style-type: none"> <li>l2_cache_miss_ratio in L2_Cache_Effectiveness</li> <li>l2_cache_miss_ratio in Miss_Ratio</li> </ul>  | <ul style="list-style-type: none"> <li>L2_Cache_Effectiveness</li> <li>Miss_Ratio</li> </ul>               | <ul style="list-style-type: none"> <li>L2_Cache</li> </ul>       |
| 0x0017, L2D_CACHE_REFILL   | <ul style="list-style-type: none"> <li>l2_cache_mпки in L2_Cache_Effectiveness</li> <li>l2_cache_mпки in MPKI</li> <li>l2_cache_miss_ratio in L2_Cache_Effectiveness</li> <li>l2_cache_miss_ratio in Miss_Ratio</li> </ul>              | <ul style="list-style-type: none"> <li>Miss_Ratio</li> <li>MPKI</li> <li>L2_Cache_Effectiveness</li> </ul> | <ul style="list-style-type: none"> <li>L2_Cache</li> </ul>       |
| 0x0018, L2D_CACHE_WB       | -   | -  | <ul style="list-style-type: none"> <li>L2_Cache</li> </ul>       |
| 0x0019, BUS_ACCESS         | -   | -  | <ul style="list-style-type: none"> <li>Bus</li> </ul>            |
| 0x001A, MEMORY_ERROR       | -   | -  | <ul style="list-style-type: none"> <li>Memory</li> </ul>         |
| 0x001B, INST_SPEC          | <ul style="list-style-type: none"> <li>load_percentage</li> <li>store_percentage</li> <li>integer_dp_percentage</li> <li>simd_percentage</li> <li>scalar_fp_percentage</li> <li>branch_percentage</li> <li>crypto_percentage</li> </ul> | <ul style="list-style-type: none"> <li>Operation_Mix</li> </ul>  | <ul style="list-style-type: none"> <li>Spec_Operation</li> </ul> |
| 0x001C, TTBR_WRITE_RETIRED | -   | -  | <ul style="list-style-type: none"> <li>Retired</li> </ul>        |
| 0x001D, BUS_CYCLES         | -   | -  | <ul style="list-style-type: none"> <li>Bus</li> </ul>            |
| 0x001E, CHAIN              | -   | -  | <ul style="list-style-type: none"> <li>Chain</li> </ul>          |
| 0x0020, L2D_CACHE_ALLOCATE | -   | -  | <ul style="list-style-type: none"> <li>L2_Cache</li> </ul>       |
| 0x0021, BR_RETIRED         | <ul style="list-style-type: none"> <li>branch_misprediction_ratio in Branch_Effectiveness</li> <li>branch_misprediction_ratio in Miss_Ratio</li> </ul>  | <ul style="list-style-type: none"> <li>Branch_Effectiveness</li> <li>Miss_Ratio</li> </ul>                 | <ul style="list-style-type: none"> <li>Retired</li> </ul>        |

| Code, Mnemonic              | Metrics  | Metric Groups  | Functional Groups  |
|-----------------------------|--|--|--|
| 0x0022, BR_MIS_PRED_RETIRED | <ul style="list-style-type: none"> <li>branch_mpki in Branch_Effectiveness</li> <li>branch_mpki in MPKI</li> <li>branch_misprediction_ratio in Branch_Effectiveness</li> <li>branch_misprediction_ratio in Miss_Ratio</li> </ul>   | <ul style="list-style-type: none"> <li>Branch_Effectiveness</li> <li>MPKI</li> <li>Miss_Ratio</li> </ul>                           | <ul style="list-style-type: none"> <li>Retired</li> </ul>  |
| 0x0023, STALL_FRONTEND      | <ul style="list-style-type: none"> <li>frontend_stalled_cycles</li> </ul>  | <ul style="list-style-type: none"> <li>Cycle_Accounting</li> </ul>   | <ul style="list-style-type: none"> <li>Stall</li> </ul>    |
| 0x0024, STALL_BACKEND       | <ul style="list-style-type: none"> <li>backend_stalled_cycles</li> </ul>   | <ul style="list-style-type: none"> <li>Cycle_Accounting</li> </ul>   | <ul style="list-style-type: none"> <li>Stall</li> </ul>    |
| 0x0025, L1D_TLB             | <ul style="list-style-type: none"> <li>dtlb_walk_ratio in DTLB_Effectiveness</li> <li>dtlb_walk_ratio in Miss_Ratio</li> <li>l1d_tlb_miss_ratio in DTLB_Effectiveness</li> <li>l1d_tlb_miss_ratio in Miss_Ratio</li> </ul>   | <ul style="list-style-type: none"> <li>DTLB_Effectiveness</li> <li>Miss_Ratio</li> </ul>   | <ul style="list-style-type: none"> <li>TLB</li> </ul>      |
| 0x0026, L1I_TLB             | <ul style="list-style-type: none"> <li>itlb_walk_ratio in ITLB_Effectiveness</li> <li>itlb_walk_ratio in Miss_Ratio</li> <li>l1i_tlb_miss_ratio in ITLB_Effectiveness</li> <li>l1i_tlb_miss_ratio in Miss_Ratio</li> </ul>   | <ul style="list-style-type: none"> <li>ITLB_Effectiveness</li> <li>Miss_Ratio</li> </ul>   | <ul style="list-style-type: none"> <li>TLB</li> </ul>      |
| 0x0029, L3D_CACHE_ALLOCATE  | -  | -  | <ul style="list-style-type: none"> <li>L3_Cache</li> </ul> |
| 0x002A, L3D_CACHE_REFILL    | -  | -  | <ul style="list-style-type: none"> <li>L3_Cache</li> </ul> |
| 0x002B, L3D_CACHE           | -  | -  | <ul style="list-style-type: none"> <li>L3_Cache</li> </ul> |
| 0x002D, L2D_TLB_REFILL      | <ul style="list-style-type: none"> <li>l2_tlb_mpki in ITLB_Effectiveness</li> <li>l2_tlb_mpki in DTLB_Effectiveness</li> <li>l2_tlb_mpki in MPKI</li> <li>l2_tlb_miss_ratio in ITLB_Effectiveness</li> <li>l2_tlb_miss_ratio in DTLB_Effectiveness</li> <li>l2_tlb_miss_ratio in Miss_Ratio</li> </ul> | <ul style="list-style-type: none"> <li>MPKI</li> <li>ITLB_Effectiveness</li> <li>DTLB_Effectiveness</li> <li>Miss_Ratio</li> </ul> | <ul style="list-style-type: none"> <li>TLB</li> </ul>      |
| 0x002F, L2D_TLB             | <ul style="list-style-type: none"> <li>l2_tlb_miss_ratio in ITLB_Effectiveness</li> <li>l2_tlb_miss_ratio in DTLB_Effectiveness</li> <li>l2_tlb_miss_ratio in Miss_Ratio</li> </ul>  | <ul style="list-style-type: none"> <li>Miss_Ratio</li> <li>DTLB_Effectiveness</li> <li>ITLB_Effectiveness</li> </ul>               | <ul style="list-style-type: none"> <li>TLB</li> </ul>      |
| 0x0031, REMOTE_ACCESS       | -  | -  | <ul style="list-style-type: none"> <li>Memory</li> </ul>   |

| Code, Mnemonic                 | Metrics   | Metric Groups  | Functional Groups   |
|--------------------------------|---|--|---|
| 0x0034, DTLB_WALK              | <ul style="list-style-type: none"> <li>dtlb_mpki in DTLB_Effectiveness</li> <li>dtlb_mpki in MPKI</li> <li>dtlb_walk_ratio in DTLB_Effectiveness</li> <li>dtlb_walk_ratio in Miss_Ratio</li> </ul>  | <ul style="list-style-type: none"> <li>DTLB_Effectiveness</li> <li>MPKI</li> <li>Miss_Ratio</li> </ul>     | <ul style="list-style-type: none"> <li>TLB</li> </ul>       |
| 0x0035, ITLB_WALK              | <ul style="list-style-type: none"> <li>itlb_mpki in ITLB_Effectiveness</li> <li>itlb_mpki in MPKI</li> <li>itlb_walk_ratio in ITLB_Effectiveness</li> <li>itlb_walk_ratio in Miss_Ratio</li> </ul>  | <ul style="list-style-type: none"> <li>Miss_Ratio</li> <li>MPKI</li> <li>ITLB_Effectiveness</li> </ul>     | <ul style="list-style-type: none"> <li>TLB</li> </ul>       |
| 0x0036, LL_CACHE_RD            | <ul style="list-style-type: none"> <li>ll_cache_read_miss_ratio in LL_Cache_Effectiveness</li> <li>ll_cache_read_miss_ratio in Miss_Ratio</li> <li>ll_cache_read_hit_ratio</li> </ul>   | <ul style="list-style-type: none"> <li>LL_Cache_Effectiveness</li> <li>Miss_Ratio</li> </ul>               | <ul style="list-style-type: none"> <li>LL_Cache</li> </ul>  |
| 0x0037, LL_CACHE_MISS_RD       | <ul style="list-style-type: none"> <li>ll_cache_read_mpki in LL_Cache_Effectiveness</li> <li>ll_cache_read_mpki in MPKI</li> <li>ll_cache_read_miss_ratio in LL_Cache_Effectiveness</li> <li>ll_cache_read_miss_ratio in Miss_Ratio</li> <li>ll_cache_read_hit_ratio</li> </ul> | <ul style="list-style-type: none"> <li>LL_Cache_Effectiveness</li> <li>MPKI</li> <li>Miss_Ratio</li> </ul> | <ul style="list-style-type: none"> <li>LL_Cache</li> </ul>  |
| 0x0040, L1D_CACHE_RD           | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x0041, L1D_CACHE_WR           | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x0042, L1D_CACHE_REFILL_RD    | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x0043, L1D_CACHE_REFILL_WR    | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x0044, L1D_CACHE_REFILL_INNER | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x0045, L1D_CACHE_REFILL_OUTER | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x0046, L1D_CACHE_WB_VICTIM    | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x0047, L1D_CACHE_WB_CLEAN     | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x0048, L1D_CACHE_INVALID      | -   | -  | <ul style="list-style-type: none"> <li>L1D_Cache</li> </ul> |
| 0x004C, L1D_TLB_REFILL_RD      | -   | -  | <ul style="list-style-type: none"> <li>TLB</li> </ul>       |
| 0x004D, L1D_TLB_REFILL_WR      | -   | -  | <ul style="list-style-type: none"> <li>TLB</li> </ul>       |
| 0x004E, L1D_TLB_RD             | -   | -  | <ul style="list-style-type: none"> <li>TLB</li> </ul>       |
| 0x004F, L1D_TLB_WR             | -   | -  | <ul style="list-style-type: none"> <li>TLB</li> </ul>       |
| 0x0050, L2D_CACHE_RD           | -   | -  | <ul style="list-style-type: none"> <li>L2_Cache</li> </ul>  |

| Code, Mnemonic              | Metrics                 | Metric Groups   | Functional Groups |
|-----------------------------|-------------------------|-----------------|-------------------|
| 0x0051, L2D_CACHE_WR        | -                       | -               | • L2_Cache        |
| 0x0052, L2D_CACHE_REFILL_RD | -                       | -               | • L2_Cache        |
| 0x0053, L2D_CACHE_REFILL_WR | -                       | -               | • L2_Cache        |
| 0x0056, L2D_CACHE_WB_VICTIM | -                       | -               | • L2_Cache        |
| 0x0057, L2D_CACHE_WB_CLEAN  | -                       | -               | • L2_Cache        |
| 0x0058, L2D_CACHE_INVALID   | -                       | -               | • L2_Cache        |
| 0x005C, L2D_TLB_REFILL_RD   | -                       | -               | • TLB             |
| 0x005D, L2D_TLB_REFILL_WR   | -                       | -               | • TLB             |
| 0x005E, L2D_TLB_RD          | -                       | -               | • TLB             |
| 0x005F, L2D_TLB_WR          | -                       | -               | • TLB             |
| 0x0060, BUS_ACCESS_RD       | -                       | -               | • Bus             |
| 0x0061, BUS_ACCESS_WR       | -                       | -               | • Bus             |
| 0x0066, MEM_ACCESS_RD       | -                       | -               | • Memory          |
| 0x0067, MEM_ACCESS_WR       | -                       | -               | • Memory          |
| 0x0068, UNALIGNED_LD_SPEC   | -                       | -               | • Spec_Operation  |
| 0x0069, UNALIGNED_ST_SPEC   | -                       | -               | • Spec_Operation  |
| 0x006A, UNALIGNED_LDST_SPEC | -                       | -               | • Spec_Operation  |
| 0x006C, LDREX_SPEC          | -                       | -               | • Spec_Operation  |
| 0x006D, STREX_PASS_SPEC     | -                       | -               | • Spec_Operation  |
| 0x006E, STREX_FAIL_SPEC     | -                       | -               | • Spec_Operation  |
| 0x006F, STREX_SPEC          | -                       | -               | • Spec_Operation  |
| 0x0070, LD_SPEC             | • load_percentage       | • Operation_Mix | • Spec_Operation  |
| 0x0071, ST_SPEC             | • store_percentage      | • Operation_Mix | • Spec_Operation  |
| 0x0073, DP_SPEC             | • integer_dp_percentage | • Operation_Mix | • Spec_Operation  |
| 0x0074, ASE_SPEC            | • simd_percentage       | • Operation_Mix | • Spec_Operation  |
| 0x0075, VFP_SPEC            | • scalar_fp_percentage  | • Operation_Mix | • Spec_Operation  |
| 0x0076, PC_WRITE_SPEC       | -                       | -               | • Spec_Operation  |
| 0x0077, CRYPTO_SPEC         | • crypto_percentage     | • Operation_Mix | • Spec_Operation  |
| 0x0078, BR_IMMED_SPEC       | • branch_percentage     | • Operation_Mix | • Spec_Operation  |
| 0x0079, BR_RETURN_SPEC      | -                       | -               | • Spec_Operation  |
| 0x007A, BR_INDIRECT_SPEC    | • branch_percentage     | • Operation_Mix | • Spec_Operation  |
| 0x007C, ISB_SPEC            | -                       | -               | • Spec_Operation  |
| 0x007D, DSB_SPEC            | -                       | -               | • Spec_Operation  |
| 0x007E, DMB_SPEC            | -                       | -               | • Spec_Operation  |
| 0x0081, EXC_UNDEF           | -                       | -               | • Exception       |
| 0x0082, EXC_SVC             | -                       | -               | • Exception       |
| 0x0083, EXC_PABORT          | -                       | -               | • Exception       |

| Code, Mnemonic                           | Metrics | Metric Groups | Functional Groups                |
|--|---------|---------------|----------------------------------|
| 0x0084, <a href="#">EXC_DABORT</a>       | -       | -             | • <a href="#">Exception</a>      |
| 0x0086, <a href="#">EXC_IRQ</a>          | -       | -             | • <a href="#">Exception</a>      |
| 0x0087, <a href="#">EXC_FIQ</a>          | -       | -             | • <a href="#">Exception</a>      |
| 0x0088, <a href="#">EXC_SMC</a>          | -       | -             | • <a href="#">Exception</a>      |
| 0x008A, <a href="#">EXC_HVC</a>          | -       | -             | • <a href="#">Exception</a>      |
| 0x008B, <a href="#">EXC_TRAP_PABORT</a>  | -       | -             | • <a href="#">Exception</a>      |
| 0x008C, <a href="#">EXC_TRAP_DABORT</a>  | -       | -             | • <a href="#">Exception</a>      |
| 0x008D, <a href="#">EXC_TRAP_OTHER</a>   | -       | -             | • <a href="#">Exception</a>      |
| 0x008E, <a href="#">EXC_TRAP_IRQ</a>     | -       | -             | • <a href="#">Exception</a>      |
| 0x008F, <a href="#">EXC_TRAP_FIQ</a>     | -       | -             | • <a href="#">Exception</a>      |
| 0x0090, <a href="#">RC_LD_SPEC</a>       | -       | -             | • <a href="#">Spec_Operation</a> |
| 0x0091, <a href="#">RC_ST_SPEC</a>       | -       | -             | • <a href="#">Spec_Operation</a> |
| 0x00A0, <a href="#">L3D_CACHE_RD</a>     | -       | -             | • <a href="#">L3_Cache</a>       |
| 0x4000, <a href="#">SAMPLE_POP</a>       | -       | -             | • <a href="#">SPE</a>            |
| 0x4001, <a href="#">SAMPLE_FEED</a>      | -       | -             | • <a href="#">SPE</a>            |
| 0x4002, <a href="#">SAMPLE_FILTRATE</a>  | -       | -             | • <a href="#">SPE</a>            |
| 0x4003, <a href="#">SAMPLE_COLLISION</a> | -       | -             | • <a href="#">SPE</a>            |



## 6. Metrics by metric group in Neoverse N1

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: 31

Metrics for Neoverse N1 are grouped into the following metric groups:

- [Cycle\\_Accounting](#), Cycle Accounting (2 metrics)
- [General](#), General (1 metrics)
- [MPKI](#), Misses Per Kilo Instructions (10 metrics)
- [Miss\\_Ratio](#), Miss Ratio (10 metrics)
- [Branch\\_Effectiveness](#), Branch Effectiveness (2 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 (7 metrics)

### 6.1 [Cycle\\_Accounting](#) metrics for Neoverse N1

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 6-1: [Cycle\\_Accounting](#) metrics summary**

| Metric                                  | Name                    | Description   |
|---|-------------------------|---|
| <a href="#">backend_stalled_cycles</a>  | Backend Stalled Cycles  | This metric is the percentage of cycles that were stalled due to resource constraints in the... |
| <a href="#">frontend_stalled_cycles</a> | 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 N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

### **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 1

### **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 1

## 6.2 General metrics for Neoverse N1

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

Summary of metrics in General:

- Total metrics: 1

**Table 6-2: General metrics summary**

| Metric              | Name                   | Description  |
|---------------------|------------------------|--|
| <a href="#">ipc</a> | Instructions Per Cycle | This metric measures the number of instructions retired per cycle. |

For a complete list of the metrics in Neoverse N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

### ipc, Instructions Per Cycle, metric

This metric measures the number of instructions retired per cycle.

#### Units

This unit is expressed as per cycle.

#### Formula

$\text{INST\_RETIRED} / \text{CPU\_CYCLES}$

#### Related telemetry artifacts

##### Events

[CPU\\_CYCLES](#)

[INST\\_RETIRED](#)

##### Metric group

[General](#)

##### Methodology

Stage 2

## 6.3 MPKI metrics for Neoverse N1

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 6-3: MPKI metrics summary**

| Metric                      | Name        | Description  |
|-----------------------------|-------------|--|
| <a href="#">branch_mпки</a> | Branch MPKI | This metric measures the number of branch mispredictions per thousand instructions executed. |

| Metric                             | Name                    | Description   |
|------------------------------------|-------------------------|---|
| <a href="#">dtlb_mpki</a>          | DTLB MPKI               | This metric measures the number of data TLB Walks per thousand instructions executed.               |
| <a href="#">itlb_mpki</a>          | ITLB MPKI               | This metric measures the number of instruction TLB Walks per thousand instructions executed.        |
| <a href="#">l1d_cache_mpki</a>     | L1D Cache MPKI          | This metric measures the number of level 1 data cache accesses missed per thousand instructions...  |
| <a href="#">l1d_tlb_mpki</a>       | L1 Data TLB MPKI        | This metric measures the number of level 1 data TLB accesses missed per thousand instructions...    |
| <a href="#">l1i_cache_mpki</a>     | L1I Cache MPKI          | This metric measures the number of level 1 instruction cache accesses missed per thousand...        |
| <a href="#">l1i_tlb_mpki</a>       | L1 Instruction TLB MPKI | This metric measures the number of level 1 instruction TLB accesses missed per thousand...          |
| <a href="#">l2_cache_mpki</a>      | L2 Cache MPKI           | This metric measures the number of level 2 unified cache accesses missed per thousand...            |
| <a href="#">l2_tlb_mpki</a>        | L2 Unified TLB MPKI     | This metric measures the number of level 2 unified TLB accesses missed per thousand instructions... |
| <a href="#">ll_cache_read_mpki</a> | 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 N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

### **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\_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

$$\text{L1I\_CACHE\_REFILL} / \text{INST\_RETIRED} * 1000$$

### Related telemetry artifacts

#### Events

[INST\\_RETIRED](#)

[L1I\\_CACHE\\_REFILL](#)

#### Metric group

[MPKI](#)

Other metric group: [L1I\\_Cache\\_Effectiveness](#)

### Methodology

Stage 2

**I1i\_tlb\_mpki, 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\_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**

$$\text{L2D\_CACHE\_REFILL} / \text{INST\_RETIRED} * 1000$$

**Related telemetry artifacts****Events**

[INST\\_RETIRED](#)

[L2D\\_CACHE\\_REFILL](#)

**Metric group**

[MPKI](#)

Other metric group: [L2\\_Cache\\_Effectiveness](#)

**Methodology**

Stage 2

**I2\_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**

$$\text{L2D\_TLB\_REFILL} / \text{INST\_RETIRED} * 1000$$

**Related telemetry artifacts****Events**

[INST\\_RETIRED](#)  
[L2D\\_TLB\\_REFILL](#)

**Metric group**

[MPKI](#)  
Other metric group: [ITLB\\_Effectiveness](#)  
Other metric group: [DTLB\\_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

## 6.4 Miss\_Ratio metrics for Neoverse N1

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 6-4: Miss\_Ratio metrics summary**

| Metric                                     | Name                          | Description  |
|--|-------------------------------|--|
| <a href="#">branch_misprediction_ratio</a> | Branch Misprediction Ratio    | This metric measures the ratio of branches mispredicted to the total number of branches...           |
| <a href="#">dttlb_walk_ratio</a>           | DTLB Walk Ratio               | This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This...   |
| <a href="#">itlb_walk_ratio</a>            | ITLB Walk Ratio               | This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB...    |
| <a href="#">l1d_cache_miss_ratio</a>       | L1D Cache Miss Ratio          | This metric measures the ratio of level 1 data cache accesses missed to the total number of level... |
| <a href="#">l1d_tlb_miss_ratio</a>         | L1 Data TLB Miss Ratio        | This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1... |
| <a href="#">l1i_cache_miss_ratio</a>       | L1I Cache Miss Ratio          | This metric measures the ratio of level 1 instruction cache accesses missed to the total number...   |
| <a href="#">l1i_tlb_miss_ratio</a>         | L1 Instruction TLB Miss Ratio | This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of...  |
| <a href="#">l2_cache_miss_ratio</a>        | L2 Cache Miss Ratio           | This metric measures the ratio of level 2 cache accesses missed to the total number of level 2...    |
| <a href="#">l2_tlb_miss_ratio</a>          | L2 Unified TLB Miss Ratio     | This metric measures the ratio of level 2 unified TLB accesses missed to the total number of...      |
| <a href="#">ll_cache_read_miss_ratio</a>   | 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 N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

### **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\\_RETIRED](#) / [BR\\_RETIRED](#)

#### **Related telemetry artifacts**

##### **Events**

[BR\\_MIS\\_PRED\\_RETIRED](#)

[BR\\_RETIRED](#)

##### **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**

$\text{DTLB\_WALK} / \text{L1D\_TLB}$

**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**

$$\frac{\text{L1I\_TLB\_REFILL}}{\text{L1I\_TLB}}$$

**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**

$$\frac{\text{L2D\_CACHE\_REFILL}}{\text{L2D\_CACHE}}$$

**Related telemetry artifacts****Events**

[L2D\\_CACHE](#)

[L2D\\_CACHE\\_REFILL](#)**Metric group**[Miss\\_Ratio](#)Other metric group: [L2\\_Cache\\_Effectiveness](#)**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**
$$\text{L2D\_TLB\_REFILL} / \text{L2D\_TLB}$$
**Related telemetry artifacts****Events**[L2D\\_TLB](#)[L2D\\_TLB\\_REFILL](#)**Metric group**[Miss\\_Ratio](#)Other metric group: [ITLB\\_Effectiveness](#)Other metric group: [DTLB\\_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**
$$\text{LL\_CACHE\_MISS\_RD} / \text{LL\_CACHE\_RD}$$
**Related telemetry artifacts****Events**[LL\\_CACHE\\_MISS\\_RD](#)[LL\\_CACHE\\_RD](#)**Metric group**[Miss\\_Ratio](#)

Other metric group: [LL\\_Cache\\_Effectiveness](#)

**Methodology**  
Stage 2

## 6.5 Branch\_Effectiveness metrics for Neoverse N1

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: 2

**Table 6-5: Branch\_Effectiveness metrics summary**

| Metric                                     | Name                       | Description  |
|--|----------------------------|--|
| <a href="#">branch_misprediction_ratio</a> | Branch Misprediction Ratio | This metric measures the ratio of branches mispredicted to the total number of branches...   |
| <a href="#">branch_mpki</a>                | Branch MPKI                | This metric measures the number of branch mispredictions per thousand instructions executed. |

For a complete list of the metrics in Neoverse N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

**[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\\_RETIRED](#)  
[BR\\_RETIRED](#)

**Metric group**

[Branch\\_Effectiveness](#)  
Other metric group: [Miss\\_Ratio](#)

**Methodology**  
Stage 2

**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**

$$\text{BR\_MIS\_PRED\_RETIRED} / \text{INST\_RETIRED} * 1000$$

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

**Related telemetry artifacts****Events**

BR\_MIS\_PRED\_RETIRED  
INST\_RETIRED

**Metric group**

Branch\_Effectiveness  
Other metric group: MPKI

**Methodology**

Stage 2

## 6.6 ITLB\_Effectiveness metrics for Neoverse N1

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 6-6: ITLB\_Effectiveness metrics summary**

| Metric             | Name                          | Description   |
|--------------------|-------------------------------|---|
| itlb_mpki          | 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_mpki       | 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_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 N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

### **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**

$$\text{ITLB\_WALK} / \text{INST\_RETIRED} * 1000$$

\*\* 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

**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 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

**`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**

`ITLB_Effectiveness`

Other metric group: `DTLB_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 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

## 6.7 DTLB\_Effectiveness metrics for Neoverse N1

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 6-7: DTLB\_Effectiveness metrics summary**

| Metric                             | Name                      | Description  |
|------------------------------------|---------------------------|--|
| <a href="#">dtlb_mpki</a>          | DTLB MPKI                 | This metric measures the number of data TLB Walks per thousand instructions executed.                |
| <a href="#">dtlb_walk_ratio</a>    | DTLB Walk Ratio           | This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This...   |
| <a href="#">l1d_tlb_miss_ratio</a> | L1 Data TLB Miss Ratio    | This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1... |
| <a href="#">l1d_tlb_mpki</a>       | L1 Data TLB MPKI          | This metric measures the number of level 1 data TLB accesses missed per thousand instructions...     |
| <a href="#">l2_tlb_miss_ratio</a>  | L2 Unified TLB Miss Ratio | This metric measures the ratio of level 2 unified TLB accesses missed to the total number of...      |
| <a href="#">l2_tlb_mpki</a>        | 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 N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

### **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**

$$\text{DTLB\_WALK} / \text{INST\_RETIRED} * 1000$$

\*\* 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**

[L1D\\_TLB\\_REFILL](#) / [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**

$$\text{L2D\_TLB\_REFILL} / \text{L2D\_TLB}$$

\*\*\* 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**

[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**

$$\text{L2D\_TLB\_REFILL} / \text{INST\_RETIRED} * 1000$$

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

**Related telemetry artifacts**

**Events**

[INST\\_RETIRED](#)  
[L2D\\_TLB\\_REFILL](#)

**Metric group**

[DTLB\\_Effectiveness](#)  
Other metric group: [ITLB\\_Effectiveness](#)  
Other metric group: [MPKI](#)

**Methodology**

Stage 2

**6.8 L1I\_Cache\_Effectiveness metrics for Neoverse N1**

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 6-8: L1I\_Cache\_Effectiveness metrics summary**

| Metric                               | Name                 | Description  |
|--------------------------------------|----------------------|--|
| <a href="#">l1i_cache_miss_ratio</a> | L1I Cache Miss Ratio | This metric measures the ratio of level 1 instruction cache accesses missed to the total number... |
| <a href="#">l1i_cache_mпки</a>       | 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 N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

**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**

$$\text{L1I\_CACHE\_REFILL} / \text{L1I\_CACHE}$$

\*\* 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\_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 mпки.

**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

## 6.9 L1D\_Cache\_Effectiveness metrics for Neoverse N1

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 6-9: L1D\_Cache\_Effectiveness metrics summary**

| Metric                               | Name                 | Description  |
|--------------------------------------|----------------------|--|
| <a href="#">l1d_cache_miss_ratio</a> | L1D Cache Miss Ratio | This metric measures the ratio of level 1 data cache accesses missed to the total number of level... |
| <a href="#">l1d_cache_mpk</a>        | 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 N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

### **`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**

[L1D\\_CACHE\\_REFILL](#) / [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_mpk`\*\***, 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

$$\text{L1D\_CACHE\_REFILL} / \text{INST\_RETIRED} * 1000$$

\*\* 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

6.10 L2\_Cache\_Effectiveness metrics for Neoverse N1

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 6-10: L2\_Cache\_Effectiveness metrics summary

| Metric                              | Name                | Description   |
|-------------------------------------|---------------------|---|
| <a href="#">l2_cache_miss_ratio</a> | L2 Cache Miss Ratio | This metric measures the ratio of level 2 cache accesses missed to the total number of level 2... |
| <a href="#">l2_cache_mпки</a>       | 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 N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

**[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

$$\text{L2D\_CACHE\_REFILL} / \text{L2D\_CACHE}$$

\*\* 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**

$$\frac{\text{L2D\_CACHE\_REFILL}}{\text{INST\_RETIRED}} * 1000$$

\*\* 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

## 6.11 LL\_Cache\_Effectiveness metrics for Neoverse N1

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 6-11: LL\_Cache\_Effectiveness metrics summary**

| Metric                                   | Name                     | Description   |
|--|--------------------------|---|
| <a href="#">ll_cache_read_hit_ratio</a>  | LL Cache Read Hit Ratio  | This metric measures the ratio of last level cache read accesses hit in the cache to the total... |
| <a href="#">ll_cache_read_miss_ratio</a> | LL Cache Read Miss Ratio | This metric measures the ratio of last level cache read accesses missed to the total number of... |
| <a href="#">ll_cache_read_mpki</a>       | 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 N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

### **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

## 6.12 Operation\_Mix metrics for Neoverse N1

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: 7

**Table 6-12: Operation\_Mix metrics summary**

| Metric                                | Name                                 | Description  |
|---------------------------------------|--------------------------------------|--|
| <a href="#">branch_percentage</a>     | Branch Operations Percentage         | This metric measures branch operations as a percentage of operations speculatively executed.         |
| <a href="#">crypto_percentage</a>     | Crypto Operations Percentage         | This metric measures crypto operations as a percentage of operations speculatively executed.         |
| <a href="#">integer_dp_percentage</a> | Integer Operations Percentage        | This metric measures scalar integer operations as a percentage of operations speculatively executed. |
| <a href="#">load_percentage</a>       | Load Operations Percentage           | This metric measures load operations as a percentage of operations speculatively executed.           |
| <a href="#">scalar_fp_percentage</a>  | Floating Point Operations Percentage | This metric measures scalar floating point operations as a percentage of operations speculatively... |
| <a href="#">simd_percentage</a>       | Advanced SIMD Operations Percentage  | This metric measures advanced SIMD operations as a percentage of total operations speculatively...   |
| <a href="#">store_percentage</a>      | Store Operations Percentage          | This metric measures store operations as a percentage of operations speculatively executed.          |

For a complete list of the metrics in Neoverse N1, see [Metrics cheat sheet for Neoverse N1](#) and [Metrics lookup table for Neoverse N1](#).

### **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**

$$(\text{BR\_IMMED\_SPEC} + \text{BR\_INDIRECT\_SPEC}) / \text{INST\_SPEC} * 100$$

#### **Related telemetry artifacts**

##### **Events**

[BR\\_IMMED\\_SPEC](#)  
[BR\\_INDIRECT\\_SPEC](#)  
[INST\\_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**
$$\text{DP\_SPEC} / \text{INST\_SPEC} * 100$$
**Related telemetry artifacts****Events**[DP\\_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**

$$\text{ST\_SPEC} / \text{INST\_SPEC} * 100$$

## Related telemetry artifacts

### Events

[INST\\_SPEC](#)

[ST\\_SPEC](#)

### Metric group

[Operation\\_Mix](#)

### Methodology

Stage 2



## 7. PMU events by functional group in Neoverse N1

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 N1 provides the following types of PMU events:

- Total implemented Common events: 110
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

PMU events for Neoverse N1 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 (2 events)
- [L2\\_Cache](#), L2 CACHE (11 events)
- [L3\\_Cache](#), L3 CACHE (4 events)
- [LL\\_Cache](#), LL CACHE (2 events)
- [Memory](#), MEMORY (5 events)
- [Retired](#), RETIRED (6 events)
- [SPE](#), SPE (4 events)
- [Spec\\_Operation](#), SPEC OPERATION (25 events)
- [Stall](#), STALL (2 events)
- [General](#), GENERAL (1 events)
- [TLB](#), TLB (16 events)

### 7.1 Bus (BUS) events for Neoverse N1

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 7-1: Bus events summary**

| Code   | Mnemonic                      | Name              | Description  |
|--------|-------------------------------|-------------------|--|
| 0x0019 | <a href="#">BUS_ACCESS</a>    | Bus access        | Counts memory transactions issued by the CPU to the external bus, including snoop requests and...    |
| 0x001D | <a href="#">BUS_CYCLES</a>    | Bus cycle         | Counts bus cycles in the CPU. Bus cycles represent a clock cycle in which a transaction could be...  |
| 0x0060 | <a href="#">BUS_ACCESS_RD</a> | Bus access, read  | Counts memory read transactions seen on the external bus. Each beat of data is counted individually. |
| 0x0061 | <a href="#">BUS_ACCESS_WR</a> | 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 N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

#### **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 N1 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 N1 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 N1 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 N1 Methodology Specification.

**Functional groups**

Bus

## 7.2 Chain (CHAIN) events for Neoverse N1

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 7-2: Chain events summary**

| Code   | Mnemonic | Name                           | Description  |
|--------|----------|--------------------------------|--|
| 0x001E | CHAIN    | Chain a pair of event counters | Counts whenever the even numbered PMU counter registers overflow. This event is used when the... |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

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

Counts whenever the even numbered PMU counter registers overflow. 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 N1 Methodology Specification.

**Functional groups**

Chain

## 7.3 Exception (EXCEPTION) events for Neoverse N1

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 7-3: 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. Eg: 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.                         |
| 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 N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

**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 N1 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. Eg: AArch64: ERET

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 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 N1 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 N1 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 N1 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 N1 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 N1 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 N1 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 N1 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 N1 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 N1 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 N1 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 N1 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 N1 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 N1 Methodology Specification.

**Functional groups**

[Exception](#)

## 7.4 L1D\_Cache (L1D CACHE) events for Neoverse N1

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 7-4: L1D\_Cache events summary**

| Code   | Mnemonic                            | Name                             | Description  |
|--------|-------------------------------------|----------------------------------|--|
| 0x0003 | <a href="#">L1D_CACHE_REFILL</a>    | Level 1 data cache refill        | Counts level 1 data cache refills caused by speculatively executed load or store operations that...  |
| 0x0004 | <a href="#">L1D_CACHE</a>           | Level 1 data cache access        | Counts level 1 data cache accesses from any load/store operations. Atomic operations that resolve... |
| 0x0015 | <a href="#">L1D_CACHE_WB</a>        | 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...  |
| 0x0040 | <a href="#">L1D_CACHE_RD</a>        | Level 1 data cache access, read  | Counts level 1 data cache accesses from any load operation. Atomic load operations that resolve...   |
| 0x0041 | <a href="#">L1D_CACHE_WR</a>        | Level 1 data cache access, write | Counts level 1 data cache accesses generated by store operations. This event also counts accesses... |
| 0x0042 | <a href="#">L1D_CACHE_REFILL_RD</a> | Level 1 data cache refill, read  | Counts level 1 data cache refills caused by speculatively executed load instructions where the...    |



| Code   | Mnemonic                               | Name  | Description   |
|--------|--|---|---|
| 0x0043 | <a href="#">L1D_CACHE_REFILL_WR</a>    | Level 1 data cache refill, write                      | Counts level 1 data cache refills caused by speculatively executed store instructions where the...  |
| 0x0044 | <a href="#">L1D_CACHE_REFILL_INNER</a> | Level 1 data cache refill, inner                      | Counts level 1 data cache refills where the cache line data came from caches inside the immediate...  |
| 0x0045 | <a href="#">L1D_CACHE_REFILL_OUTER</a> | Level 1 data cache refill, outer                      | Counts level 1 data cache refills for which the cache line data came from outside the immediate...  |
| 0x0046 | <a href="#">L1D_CACHE_WB_VICTIM</a>    | Level 1 data cache write-back, victim                 | Counts dirty cache line evictions from the level 1 data cache caused by a new cache line...   |
| 0x0047 | <a href="#">L1D_CACHE_WB_CLEAN</a>     | Level 1 data cache write-back, cleaning and coherency | Counts write-backs from the level 1 data cache that are a result of a coherency operation made by...  |
| 0x0048 | <a href="#">L1D_CACHE_INVAL</a>        | 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"> <li>Cache...</li> </ul> |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

### 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. This event does not count cache line allocations from preload instructions or from hardware cache prefetching.

#### 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 N1 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 N1 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 N1 Methodology Specification.

**Functional groups**

[L1D\\_Cache](#)

**0x0042 L1D\_CACHE\_REFILL\_RD, Level 1 data cache refill, read, event**

Counts level 1 data cache refills caused by speculatively executed load instructions where the memory read operation misses in the level 1 data cache. This event only counts one event per cache line.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**

[L1D\\_Cache](#)

**0x0043 L1D\_CACHE\_REFILL\_WR, Level 1 data cache refill, write, event**

Counts level 1 data cache refills caused by speculatively executed store instructions where the memory write operation misses in the level 1 data cache. This event only counts one event per cache line.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 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 N1 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 N1 Methodology Specification.

**Functional groups**

[L1D\\_Cache](#)

**0x0046 L1D\_CACHE\_WB\_VICTIM, Level 1 data cache write-back, victim, event**

Counts dirty cache line evictions from the level 1 data cache caused by a new cache line allocation. This event does not count evictions caused by cache maintenance operations.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**

[L1D\\_Cache](#)

**0x0047 L1D\_CACHE\_WB\_CLEAN, Level 1 data cache write-back, cleaning and coherency, event**

Counts write-backs from the level 1 data cache that are a result of a coherency operation made by another CPU. Event count includes cache maintenance operations.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 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 N1 Methodology Specification.

**Functional groups**

[L1D\\_Cache](#)

## 7.5 L1I\_Cache (L1I CACHE) events for Neoverse N1

L1 instruction cache related events.

Summary of events in L1I\_Cache:

- Total implemented Common events: 2
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0

- ETE Only events : 0

**Table 7-5: L1I\_Cache events summary**

| Code   | Mnemonic                         | Name                             | Description  |
|--------|----------------------------------|----------------------------------|--|
| 0x0001 | <a href="#">L1I_CACHE_REFILL</a> | Level 1 instruction cache refill | Counts cache line refills in the level 1 instruction cache caused by a missed instruction fetch....  |
| 0x0014 | <a href="#">L1I_CACHE</a>        | Level 1 instruction cache access | Counts instruction fetches which access the level 1 instruction cache. Instruction cache accesses... |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

#### **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](#)

## 7.6 L2\_Cache (L2 CACHE) events for Neoverse N1

L2 unified cache related events.

Summary of events in L2\_Cache:

- Total implemented Common events: 11
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

**Table 7-6: L2\_Cache events summary**

| Code   | Mnemonic            | Name  | Description  |
|--------|---------------------|---|--|
| 0x0016 | L2D_CACHE           | Level 2 data cache access                             | Counts level 2 cache accesses. Level 2 cache is a unified cache for data and instruction...          |
| 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 allocates that do not fetch data from outside the level 2 data or...       |
| 0x0050 | L2D_CACHE_RD        | Level 2 data cache access, read                       | Counts level 2 cache accesses due to memory read operations. Level 2 cache is a unified cache for... |
| 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...  |
| 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:<br>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...       |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

### 0x0016 L2D\_CACHE, Level 2 data cache access, event

Counts level 2 cache accesses. Level 2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the first level caches 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 caches 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**[Miss\\_Ratio](#)[MPKI](#)[L2\\_Cache\\_Effectiveness](#)**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 N1 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 N1 Methodology Specification.

**Functional groups**

[L2\\_Cache](#)

**0x0050 L2D\_CACHE\_RD, Level 2 data cache access, read, event**

Counts level 2 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 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 N1 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 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 N1 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 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 N1 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 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 N1 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 N1 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 N1 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 N1 Methodology Specification.

**Functional groups**

[L2\\_Cache](#)

## 7.7 L3\_Cache (L3 CACHE) events for Neoverse N1

L3 unified cache related events.

Summary of events in L3\_Cache:

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

**Table 7-7: L3\_Cache events summary**

| Code   | Mnemonic                           | Name   | Description  |
|--------|------------------------------------|--|--|
| 0x0029 | <a href="#">L3D_CACHE_ALLOCATE</a> | 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 | <a href="#">L3D_CACHE_REFILL</a>   | Level 3 data cache refill                    | Counts level 3 accesses that receive data from outside the L3 cache.                             |
| 0x002B | <a href="#">L3D_CACHE</a>          | Level 3 data cache access                    | Counts level 3 cache accesses. Level 3 cache is a unified cache for data and instruction...      |
| 0x00A0 | <a href="#">L3D_CACHE_RD</a>       | Level 3 data cache access, read              | Counts level 3 cache accesses caused by any memory read operation. Level 3 cache is a unified... |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

**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 N1 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 N1 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 N1 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 N1 Methodology Specification.

**Functional groups**

[L3\\_Cache](#)

## 7.8 LL\_Cache (LL CACHE) events for Neoverse N1

Last Level Cache related events.

Summary of events in LL\_Cache:

- Total implemented Common events: 2
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

**Table 7-8: LL\_Cache events summary**

| Code   | Mnemonic                         | Name                          | Description   |
|--------|----------------------------------|-------------------------------|---|
| 0x0036 | <a href="#">LL_CACHE_RD</a>      | Last level cache access, read | Counts read transactions that were returned from outside the core cluster. This event counts for... |
| 0x0037 | <a href="#">LL_CACHE_MISS_RD</a> | Last level cache miss, read   | Counts read transactions that were returned from outside the core cluster but missed in the...      |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

### **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](#)

## 7.9 Memory (MEMORY) events for Neoverse N1

Memory system related events.

Summary of events in Memory:

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

**Table 7-9: Memory events summary**

| Code   | Mnemonic                      | Name  | Description  |
|--------|-------------------------------|---|--|
| 0x0013 | <a href="#">MEM_ACCESS</a>    | Data memory access                                | Counts memory accesses issued by the CPU load store unit, where those accesses are issued due to...  |
| 0x001A | <a href="#">MEMORY_ERROR</a>  | Local memory error                                | Counts any detected correctable or uncorrectable physical memory errors (ECC or parity) in...        |
| 0x0031 | <a href="#">REMOTE_ACCESS</a> | 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...   |
| 0x0066 | <a href="#">MEM_ACCESS_RD</a> | Data memory access, read                          | Counts memory accesses issued by the CPU due to load operations. The event counts any memory load... |
| 0x0067 | <a href="#">MEM_ACCESS_WR</a> | Data memory access, write                         | Counts memory accesses issued by the CPU due to store operations. The event counts any memory...     |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

### 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 N1 Methodology Specification.

#### Functional groups

[Memory](#)

### 0x001A MEMORY\_ERROR, Local memory error, event

Counts any detected correctable or uncorrectable physical memory errors (ECC or parity) in protected CPUs RAMs. On the core, this event counts errors in the caches (including data and tag rams). Any detected memory error (from either a speculative and abandoned access, or an architecturally executed access) is counted. Note that errors are only detected when the actual protected memory is accessed by an operation.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 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 N1 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 N1 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 N1 Methodology Specification.

**Functional groups**

Memory

## 7.10 Retired (RETIRED) events for Neoverse N1

Retired instruction and operation events.

Summary of events in Retired:

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

**Table 7-10: 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...    |
| 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 register, which usually contain the...    |
| 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.         |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

### 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 N1 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 ITLB\\_Effectiveness](#)  
[l2\\_tlb\\_mpki in DTLB\\_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**

[LL\\_Cache\\_Effectiveness](#)  
[MPKI](#)  
[L1D\\_Cache\\_Effectiveness](#)  
[L2\\_Cache\\_Effectiveness](#)  
[Branch\\_Effectiveness](#)  
[General](#)  
[ITLB\\_Effectiveness](#)  
[L1I\\_Cache\\_Effectiveness](#)  
[DTLB\\_Effectiveness](#)



**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 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 N1 Methodology Specification.

**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 N1 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.

**Related telemetry artifacts****Metrics**[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\_RETIRED, Branch instruction architecturally executed, mispredicted, event**

Counts branches counted by BR\_RETIRED 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](#)

## 7.11 SPE (SPE) events for Neoverse N1

SPE related events.

Summary of events in SPE:

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

**Table 7-11: SPE events summary**

| Code   | Mnemonic                         | Name  | Description  |
|--------|----------------------------------|---|--|
| 0x4000 | <a href="#">SAMPLE_POP</a>       | Statistical Profiling sample population                         | Counts statistical profiling sample population, the count of all operations that could be sampled... |
| 0x4001 | <a href="#">SAMPLE_FEED</a>      | Statistical Profiling sample taken                              | Counts statistical profiling samples taken for sampling.   |
| 0x4002 | <a href="#">SAMPLE_FILTRATE</a>  | Statistical Profiling sample taken and not removed by filtering | Counts statistical profiling samples taken which are not removed by filtering.                       |
| 0x4003 | <a href="#">SAMPLE_COLLISION</a> | Statistical Profiling sample collided with previous sample      | Counts statistical profiling samples that have collided with a previous sample and so therefore...   |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

### 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 N1 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 N1 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 N1 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 N1 Methodology Specification.

**Functional groups**

[SPE](#)

## 7.12 Spec\_Operation (SPEC OPERATION) events for Neoverse N1

Speculatively executed operations related events.

Summary of events in Spec\_Operation:

- Total implemented Common events: 25

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

**Table 7-12: 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 branches speculatively executed and were predicted right.                                     |
| 0x001B | INST_SPEC           | Operation speculatively executed   | Counts operations that have been speculatively executed.   |
| 0x0068 | UNALIGNED_LD_SPEC   | Unaligned access, read   | Counts unaligned memory read operations issued by the CPU. This event counts unaligned accesses...   |
| 0x0069 | UNALIGNED_ST_SPEC   | Unaligned access, write  | Counts unaligned memory write operations issued by the CPU. This event counts unaligned accesses...  |
| 0x006A | UNALIGNED_LDST_SPEC | Unaligned access   | Counts unaligned memory operations issued by the CPU. This event counts unaligned accesses (as...    |
| 0x006C | LDREX_SPEC          | Exclusive operation speculatively executed, Load-Exclusive               | Counts Load-Exclusive operations that have been speculatively executed. Eg: LDREX, LDX               |
| 0x006D | STREX_PASS_SPEC     | Exclusive operation speculatively executed, Store-Exclusive pass         | Counts store-exclusive operations that have been speculatively executed and have successfully...     |
| 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.        |
| 0x0078 | BR_IMMED_SPEC       | Branch speculatively executed, immediate branch                          | Counts immediate branch operations which are speculatively executed.                                 |
| 0x0079 | BR_RETURN_SPEC      | Branch speculatively executed, procedure return                          | Counts procedure return operations (RET) which are speculatively executed.                           |
| 0x007A | BR_INDIRECT_SPEC    | Branch speculatively executed, indirect branch                           | Counts indirect branch operations including procedure returns, which are speculatively executed....  |
| 0x007C | ISB_SPEC            | Barrier speculatively executed, ISB                                      | Counts ISB operations that are executed.   |

| Code   | Mnemonic                   | Name  | Description  |
|--------|----------------------------|---|--|
| 0x007D | <a href="#">DSB_SPEC</a>   | Barrier speculatively executed, DSB                                 | Counts DSB operations that are speculatively issued to Load/Store unit in the CPU.                   |
| 0x007E | <a href="#">DMB_SPEC</a>   | Barrier speculatively executed, DMB                                 | Counts DMB operations that are speculatively issued to the Load/Store unit in the CPU. This event... |
| 0x0090 | <a href="#">RC_LD_SPEC</a> | Release consistency operation speculatively executed, Load-Acquire  | Counts any load acquire operations that are speculatively executed. Eg: LDAR, LDARH, LDARB           |
| 0x0091 | <a href="#">RC_ST_SPEC</a> | Release consistency operation speculatively executed, Store-Release | Counts any store release operations that are speculatively executed. Eg: STLR, STLRH, STLRB'         |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

### **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 N1 Methodology Specification.

#### **Functional groups**

[Spec\\_Operation](#)

### **0x0012 BR\_PRED, Predictable branch instruction speculatively executed, event**

Counts branches speculatively executed and were predicted right.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 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](#)  
[branch\\_percentage](#)  
[crypto\\_percentage](#)

**Metric groups**[Operation\\_Mix](#)**Functional groups**[Spec\\_Operation](#)**0x0068 UNALIGNED\_LD\_SPEC, Unaligned access, read, event**

Counts unaligned memory read operations issued by the CPU. This event counts unaligned accesses (as defined by the actual instruction), even if they are subsequently issued as multiple aligned accesses. The event does not count preload operations (PLD, PLI).

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**[Spec\\_Operation](#)**0x0069 UNALIGNED\_ST\_SPEC, Unaligned access, write, event**

Counts unaligned memory write operations issued by the CPU. This event counts unaligned accesses (as defined by the actual instruction), even if they are subsequently issued as multiple aligned accesses.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**[Spec\\_Operation](#)**0x006A UNALIGNED\_LDST\_SPEC, Unaligned access, event**

Counts unaligned memory operations issued by the CPU. This event counts unaligned accesses (as defined by the actual instruction), even if they are subsequently issued as multiple aligned accesses.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**[Spec\\_Operation](#)**0x006c LDREX\_SPEC, Exclusive operation speculatively executed, Load-Exclusive, event**

Counts Load-Exclusive operations that have been speculatively executed. Eg: LDREX, LDX

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**[Spec\\_Operation](#)

**0x006D STREX\_PASS\_SPEC, Exclusive operation speculatively executed, Store-Exclusive pass, event**

Counts store-exclusive operations that have been speculatively executed and have 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 N1 Methodology Specification.

**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 N1 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 N1 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**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**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](#)**0x0078 BR\_IMMED\_SPEC, Branch speculatively executed, immediate branch, event**

Counts immediate branch operations which are speculatively executed.

**Related telemetry artifacts****Metrics**[branch\\_percentage](#)**Metric groups**[Operation\\_Mix](#)**Functional groups**[Spec\\_Operation](#)**0x0079 BR\_RETURN\_SPEC, Branch speculatively executed, procedure return, event**

Counts procedure return operations (RET) which are speculatively executed.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**

[Spec\\_Operation](#)

**0x007A BR\_INDIRECT\_SPEC, Branch speculatively executed, indirect branch, event**

Counts indirect branch operations including procedure returns, which are speculatively executed. This includes operations that force a software change of the PC, other than exception-generating operations. Eg: BR Xn, RET

**Related telemetry artifacts****Metrics**

[branch\\_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**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**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**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**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**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**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. Eg: LDAR, LDARH, LDARB

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 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. Eg: STLR, STLRH, STLRB'

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**[Spec\\_Operation](#)

## 7.13 Stall (STALL) events for Neoverse N1

Stall related events.

Summary of events in Stall:

- Total implemented Common events: 2
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

**Table 7-13: Stall events summary**

| Code   | Mnemonic                       | Name  | Description  |
|--------|--------------------------------|---|--|
| 0x0023 | <a href="#">STALL_FRONTEND</a> | 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 | <a href="#">STALL_BACKEND</a>  | 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... |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

**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. All the frontend slots were empty during the cycle when this event counts.

**Related telemetry artifacts**

- Metrics**
  - frontend\_stalled\_cycles
- Metric groups**
  - Cycle\_Accounting
- 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
- Metric groups**
  - Cycle\_Accounting
- Functional groups**
  - Stall

## 7.14 General (GENERAL) events for Neoverse N1

General CPU related events.

Summary of events in General:

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

**Table 7-14: 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... |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

### 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](#)

[ipc](#)

##### Metric groups

[Cycle\\_Accounting](#)

[General](#)

##### Functional groups

[General](#)

## 7.15 TLB (TLB) events for Neoverse N1

TLB and MMU related events.

Summary of events in TLB:

- Total implemented Common events: 16
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

**Table 7-15: TLB events summary**

| Code   | Mnemonic                       | Name                           | Description  |
|--------|--------------------------------|--------------------------------|--|
| 0x0002 | <a href="#">L1I_TLB_REFILL</a> | Level 1 instruction TLB refill | Counts level 1 instruction TLB refills from any Instruction fetch. If there are multiple misses...   |
| 0x0005 | <a href="#">L1D_TLB_REFILL</a> | Level 1 data TLB refill        | Counts level 1 data TLB accesses that resulted in TLB refills. If there are multiple misses in...    |
| 0x0025 | <a href="#">L1D_TLB</a>        | Level 1 data TLB access        | Counts level 1 data TLB accesses caused by any memory load or store operation. Note that load or...  |
| 0x0026 | <a href="#">L1I_TLB</a>        | Level 1 instruction TLB access | Counts level 1 instruction TLB accesses, whether the access hits or misses in the TLB. This event... |
| 0x002D | <a href="#">L2D_TLB_REFILL</a> | Level 2 data TLB refill        | Counts level 2 TLB refills caused by memory operations from both data and instruction fetch,...      |
| 0x002F | <a href="#">L2D_TLB</a>        | Level 2 data TLB access        | Counts level 2 TLB accesses except those caused by TLB maintenance operations.                       |

| Code   | Mnemonic                          | Name  | Description  |
|--------|-----------------------------------|---|--|
| 0x0034 | <a href="#">DTLB_WALK</a>         | 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 | <a href="#">ITLB_WALK</a>         | 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...           |
| 0x004C | <a href="#">L1D_TLB_REFILL_RD</a> | Level 1 data TLB refill, read                                   | Counts level 1 data TLB refills caused by memory read operations. If there are multiple misses in... |
| 0x004D | <a href="#">L1D_TLB_REFILL_WR</a> | Level 1 data TLB refill, write                                  | Counts level 1 data TLB refills caused by data side memory write operations. If there are...         |
| 0x004E | <a href="#">L1D_TLB_RD</a>        | Level 1 data TLB access, read                                   | Counts level 1 data TLB accesses caused by memory read operations. This event counts whether the...  |
| 0x004F | <a href="#">L1D_TLB_WR</a>        | Level 1 data TLB access, write                                  | Counts any L1 data side TLB accesses caused by memory write operations. This event counts whether... |
| 0x005C | <a href="#">L2D_TLB_REFILL_RD</a> | Level 2 data TLB refill, read                                   | Counts level 2 TLB refills caused by memory read operations from both data and instruction fetch...  |
| 0x005D | <a href="#">L2D_TLB_REFILL_WR</a> | Level 2 data TLB refill, write                                  | Counts level 2 TLB refills caused by memory write operations from both data and instruction fetch... |
| 0x005E | <a href="#">L2D_TLB_RD</a>        | Level 2 data TLB access, read                                   | Counts level 2 TLB accesses caused by memory read operations from both data and instruction fetch... |
| 0x005F | <a href="#">L2D_TLB_WR</a>        | Level 2 data TLB access, write                                  | Counts level 2 TLB accesses caused by memory write operations from both data and instruction...      |

For a complete list of the events in Neoverse N1, see [PMU events cheat sheet for Neoverse N1](#) and [PMU events lookup table for Neoverse N1](#).

### **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**

[Miss\\_Ratio](#)  
[MPKI](#)  
[ITLB\\_Effectiveness](#)

##### **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 ITLB\_Effectiveness  
l2\_tlb\_mpki in DTLB\_Effectiveness  
l2\_tlb\_mpki in MPKI  
l2\_tlb\_miss\_ratio in ITLB\_Effectiveness  
l2\_tlb\_miss\_ratio in DTLB\_Effectiveness  
l2\_tlb\_miss\_ratio in Miss\_Ratio

**Metric groups**

MPKI  
ITLB\_Effectiveness  
DTLB\_Effectiveness  
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 ITLB\_Effectiveness  
l2\_tlb\_miss\_ratio in DTLB\_Effectiveness  
l2\_tlb\_miss\_ratio in Miss\_Ratio

**Metric groups**

Miss\_Ratio



[DTLB\\_Effectiveness](#)[ITLB\\_Effectiveness](#)**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**[Miss\\_Ratio](#)[MPKI](#)

## ITLB\_Effectiveness

### Functional groups

#### TLB

#### **0x004C L1D\_TLB\_REFILL\_RD, Level 1 data TLB refill, read, event**

Counts level 1 data TLB refills caused by memory read operations. 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 Address Translation (AT) instruction.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

#### **Functional groups**

#### TLB

#### **0x004D L1D\_TLB\_REFILL\_WR, Level 1 data TLB refill, write, event**

Counts level 1 data TLB refills caused by data side memory write operations. 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 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 with an access from an Address Translation (AT) instruction.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

#### **Functional groups**

#### TLB

#### **0x004E L1D\_TLB\_RD, Level 1 data TLB access, read, event**

Counts level 1 data TLB accesses caused by memory read operations. This event counts whether the access hits or misses in the TLB. This event does not count TLB maintenance operations.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

#### **Functional groups**

#### TLB

**0x004F L1D\_TLB\_WR, Level 1 data TLB access, write, event**

Counts any L1 data side TLB accesses caused by memory write operations. This event counts whether the access hits or misses in the TLB. This event does not count TLB maintenance operations.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**

TLB

**0x005C L2D\_TLB\_REFILL\_RD, Level 2 data TLB refill, read, event**

Counts level 2 TLB refills caused by memory read operations from both data and instruction fetch except for those caused by TLB maintenance operations or hardware prefetches.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**

TLB

**0x005D L2D\_TLB\_REFILL\_WR, Level 2 data TLB refill, write, event**

Counts level 2 TLB refills caused by memory write operations from both data and instruction fetch except for those caused by TLB maintenance operations.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**

TLB

**0x005E L2D\_TLB\_RD, Level 2 data TLB access, read, event**

Counts level 2 TLB accesses caused by memory read operations from both data and instruction fetch except for those caused by TLB maintenance operations.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**

TLB

**0x005F L2D\_TLB\_WR, Level 2 data TLB access, write, event**

Counts level 2 TLB accesses caused by memory write operations from both data and instruction fetch except for those caused by TLB maintenance operations.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the Neoverse N1 Methodology Specification.

**Functional groups**

[TLB](#)