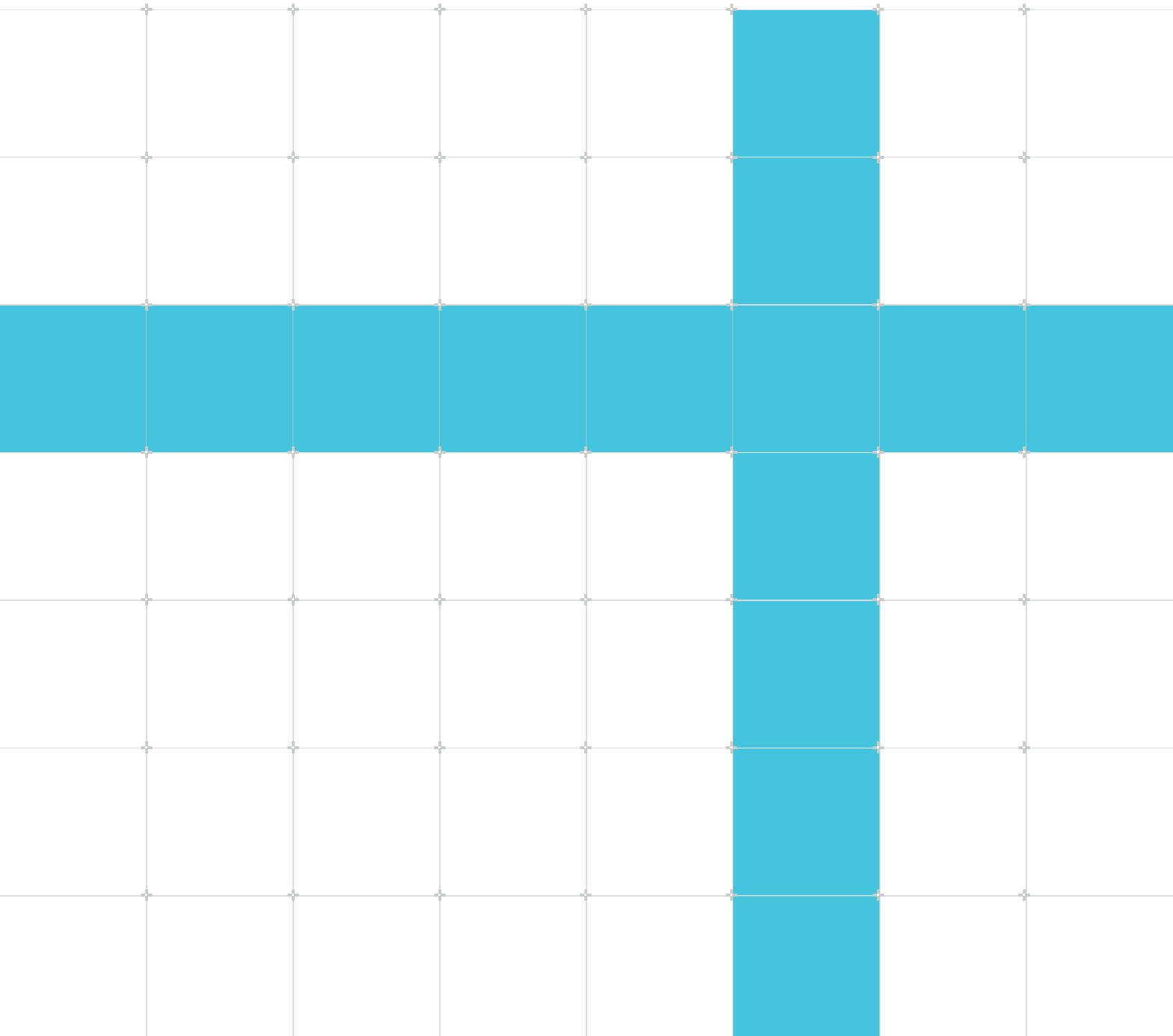
Arm® C1-Nano Core

Telemetry Specification

Non-Confidential

Issue 02
109818_0200_02_en



Arm® C1-Nano Core Telemetry Specification

This document (109818_0200_02_en) was issued on 2025-09-10. There might be a later issue at https://developer.arm.com/documentation/109818

See also: Proprietary notice | Product and document information | Useful resources

Start reading

If you prefer, you can skip to the start of the content.

Intended audience

This specification is useful for engineers to collect and analyze Arm® C1-Nano core telemetry data to gain insights about a system's performance. Architects and system designers can also use it for resource characterization and platform tuning.

Inclusive language commitment

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

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

Feedback

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

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

Contents

1. Overview of the C1-Nano Telemetry methodology

The *Arm® C1-Nano Telemetry Specification* describes the Topdown methodology, derived metrics, and Performance Monitoring Unit (PMU) events supported by the Arm C1-Nano unit.

The Arm® C1-Nano unit is also known as the processor or co-processor if an Arm® C1-SME2 unit is included in the configuration.

Note

This specification is applicable to all releases of the product. The C1-SME2 unit (previously known as the CME unit) is also referred to as the SME2 unit throughout this specification. However, some instances of the term CME remain in this specification.

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 C1-Nano 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 product 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.

1.1 Documentation and resources

Arm products include a set of documents.

The documentation and resources for C1-Nano consist of:

- *Arm® Telemetry on Arm Developer*
- *Arm® C1-Nano Core Technical Reference Manual*
- *Arm® C1-Scalable Matrix Extension 2 Telemetry Specification*

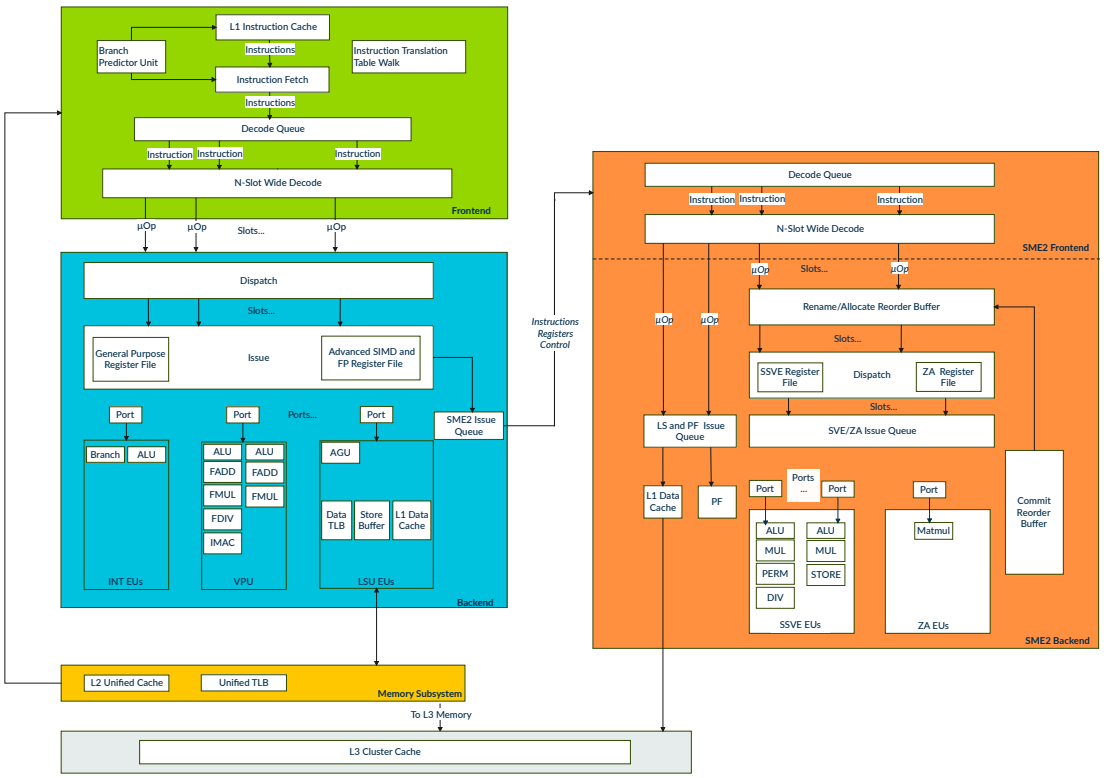
2. C1-Nano microarchitecture and its telemetry features

C1-Nano is a superscalar in-order processor, which issues and executes instructions in program order.

A C1-Nano core forms part of a C1-Nano complex, which can include up to two cores. The cores in a complex share an L2 cache, L2 Translation Lookaside Buffer (TLB), and Vector Processing Unit (VPU).

The following figure shows the microarchitecture details of C1-Nano.

Figure 2-1: C1-Nano core microarchitecture



The frontend of the core pipeline is comprised of the instruction fetch and decode units. The frontend also includes a branch predictor unit that predicts branch target addresses and direction, and fetches instructions ahead of the pipeline. This unit helps to hide latencies caused by control flow bubbles in the pipeline.

The fetch unit can fetch multiple instructions for each cycle, 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 available decode slots. The decode unit decomposes the Arm

architecture instructions into micro-operations, also known as micro-ops or µops. This unit decodes more than one micro-operation for each cycle. These micro-ops are then fed to the dispatch logic and issue logic in the backend of the core pipeline.

The dispatch unit in the backend can dispatch a bundle of up to "N" operations per cycle, which is defined by:

- The issue width of the core
- The highest Intructions Per Cycle (IPC) achievable by the core

The operations in the bundle are selected for execution in lockstep after being checked for structural or operand hazards that prevent parallel execution. Advanced forwarding techniques in the microarchitecture maximize parallelism in each bundle and minimize pipeline stalls due to operand dependencies.

The issue logic in the backend simultaneously issues the operations in the bundle to the execution ports. Issuing occurs as soon as all dependencies and structural hazards for all operations in the bundle have been resolved.

Each execution port supports different categories of operations based on the execution units (EUs) that they are connected to. When an operation is stalled in a cycle, other operations in the same bundle can also stall. These stalls can be a major cause of performance bottleneck in this processor.

The typical operation types supported by different execution units (EU) are:

- Integer EU, executes branches and arithmetic instructions
- Floating-Point Unit (FPU) and Single Instruction Multiple Data (SIMD) EU, executes the floating-point instruction and vector instructions
- Load Store Unit (LSU), executes load, store, and atomic instructions
- Scalable Matrix Extension 2 (SME2), executes scalable matrix (SME) instructions

The SME2 co-processor is a shared unit that implements SME and SME2 instructions. It is implemented inside an Arm® C1-DSU (DSU) cluster. The SME2 co-processor is shared between all the cores and is accessible through this DSU cluster. The DSU behaves as a full interconnect with shared L3 support and full coherency between the cores and the SME2 co-processor. CPU performance for SME2 workloads may be impacted by CPU managed resources or by SME2 managed resources. This specification describes the methodology to analyze CPU performance that is related to SME2 unit bottlenecks due to CPU resources. For more information about the performance analysis methodology for the resources that are managed by the SME2 co-processor, see *Arm® C1-Scalable Matrix Extension 2 Telemetry Specification*.

The Load Store Unit controls the data flow between the caches and to memory.

The memory subsystem of the core handles the execution of load and store operations which rely heavily on the memory hierarchy levels.

C1-Nano has dedicated, unique L1 instruction and data caches for each core in the complex. The L2 cache is a unified cache that is shared between cores in a complex and used for both instruction

and data. The caches are set associative and the sizes are configurable for each implementation. The cache line size for this core is 64 bytes. The L2 cache of a complex connects to the DynamIQ™ Shared Unit (DSU) cluster logic and an optional L3 cluster cache.

C1-Nano system configurations

C1-Nano connects to the compute cluster through the DSU which supports a shared L3 cache across all cores in the cluster. The DSU also supports a snoop filter for coherency management of shared data between the processors. It is possible for the platform to support other caches downstream in the system interconnects.

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 C1-Nano

The C1-Nano implements version 3.8 of the Performance Monitors Extension, FEAT_PMUv3p8, and Arm version 8.8 debug architecture, FEAT_Debugv8p8.

For more information, see *Arm® Architecture Reference Manual for A-profile architecture*.

The C1-Nano PMU can support 6 or 31 programmable performance counters, depending on your configuration, and one fixed function counter to count CPU cycles.

3. CPU performance analysis methodology

The Arm Topdown methodology for the C1-Nano enables you to use PMU events, metrics, and metric groups to identify potential bottlenecks that could negatively impact the performance of the core in your design.

The methodology is conducted in two stages.

Stage 1: Topdown analysis

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

Stage 2: Microarchitecture Exploration

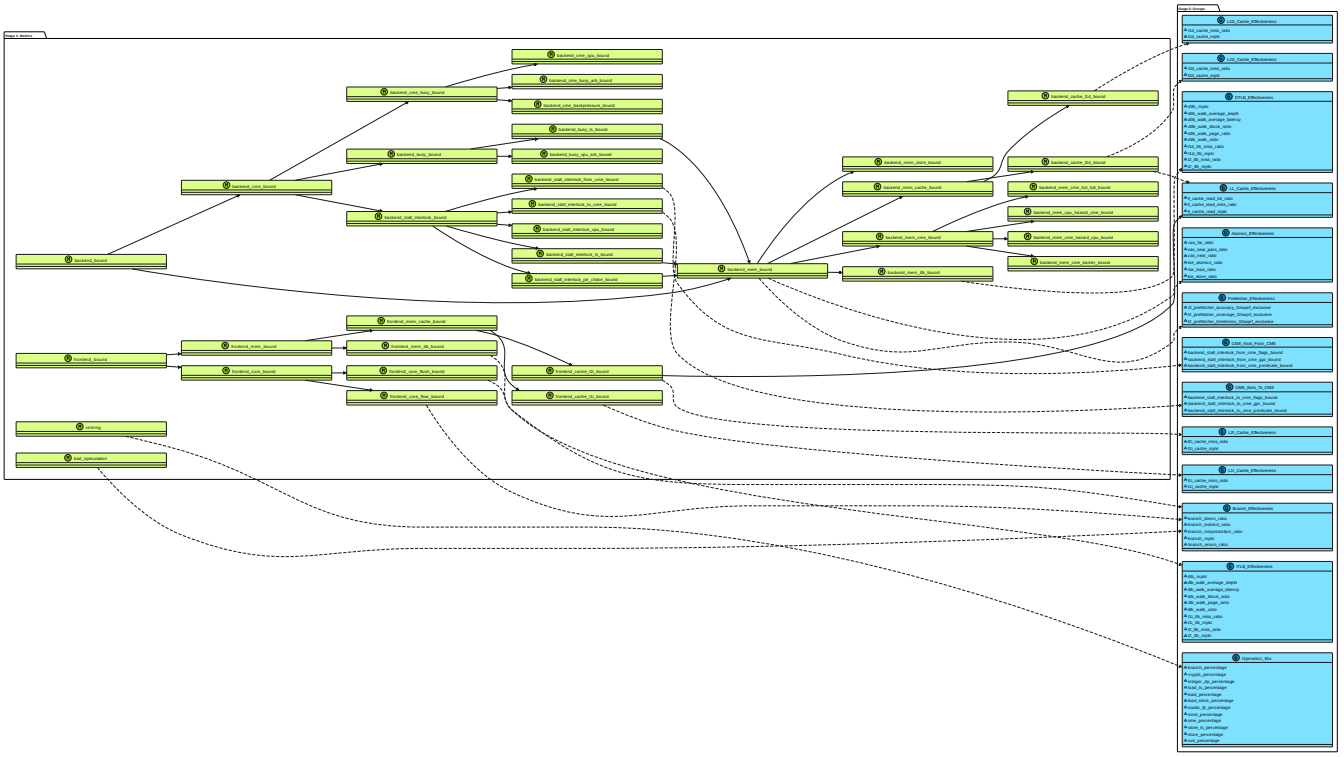
The second stage is to conduct microarchitecture exploration to further analyze bottlenecked CPU resources, based on the Stage 1 findings. Stage 2 uses a set of CPU resource-effectiveness metrics. For more information, see Stage 2: Microarchitecture Exploration.

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

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

The following figure gives an overview of the Topdown methodology tree for C1-Nano. It shows Stage 1 metrics, and Stage 2 metric groups and metrics. Stage 1 covers the stall-related metrics for Topdown analysis for bottleneck identification. Stage 2 covers the microarchitecture exploration metric groups for root cause analysis.

Figure 3-1: Topdown methodology overview for C1-Nano



Note

The industry-standard metrics Misses Per Kilo Instructions (MPKI) and Miss Ratios are metric groups that are defined in Stage 2, but they are not included in the hierarchical pipeline stall analysis of Stage 1, as these are a standard set of metrics recommended for analysis and characterization rather than hierarchical analysis specified as topdown.

3.1 Stage 1: Topdown analysis

The objective of the Topdown analysis is to identify potential bottlenecks in the key blocks of the processor.

Each block can be further broken down to units and subunits within the block that all have different performance characteristics. For example, control flow and data flow issues can be fixed differently in software after root cause analysis.

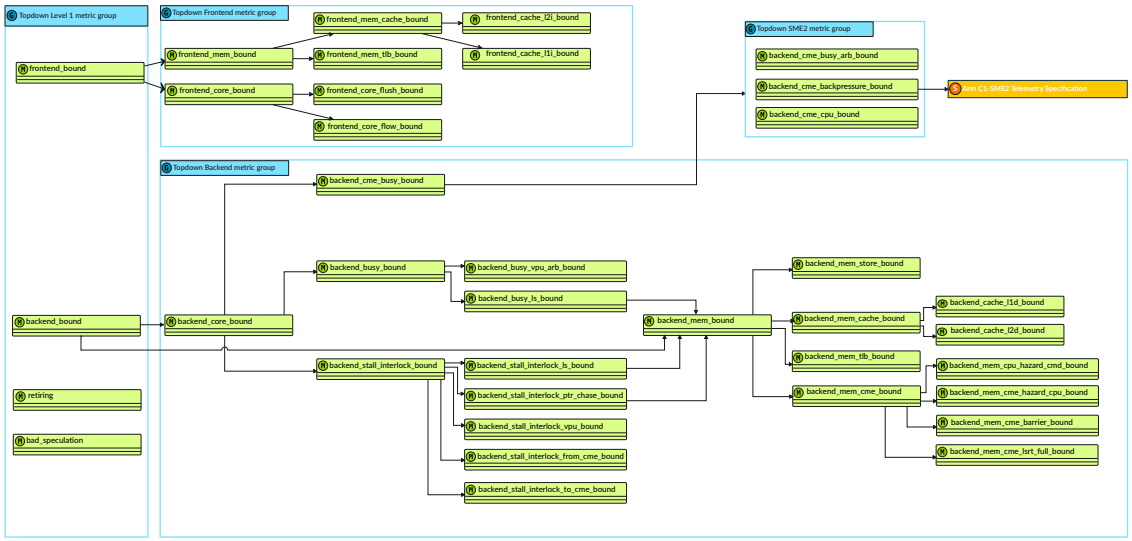
Results from a Topdown analysis can indicate:

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

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

The following figure shows the metric groups and metrics in the Stage 1 Topdown methodology tree for C1-Nano, which supports up to four levels of hierarchical pipeline stall accounting.

Figure 3-2: C1-Nano Topdown methodology Stage 1 overview



C1-Nano has four Stage 1 metric groups.

Metric Group: Topdown Level 1

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

For more information about the metrics in this group and the associated formulas and events, see Topdown_L1.

Metric Group: Topdown Frontend

The Topdown Frontend metric group contains a set of metrics to analyze a frontend bound workload.

For more information about the metrics in this group and the associated formulas and events, see Topdown_Frontend.

Metric Group: Topdown Backend

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

For more information about the metrics in this group and the associated formulas and events, see Topdown_Backend.

Metric Group: Topdown SME2

The Topdown SME2 metric group contains a set of metrics to analyze a SME2 bound workload.

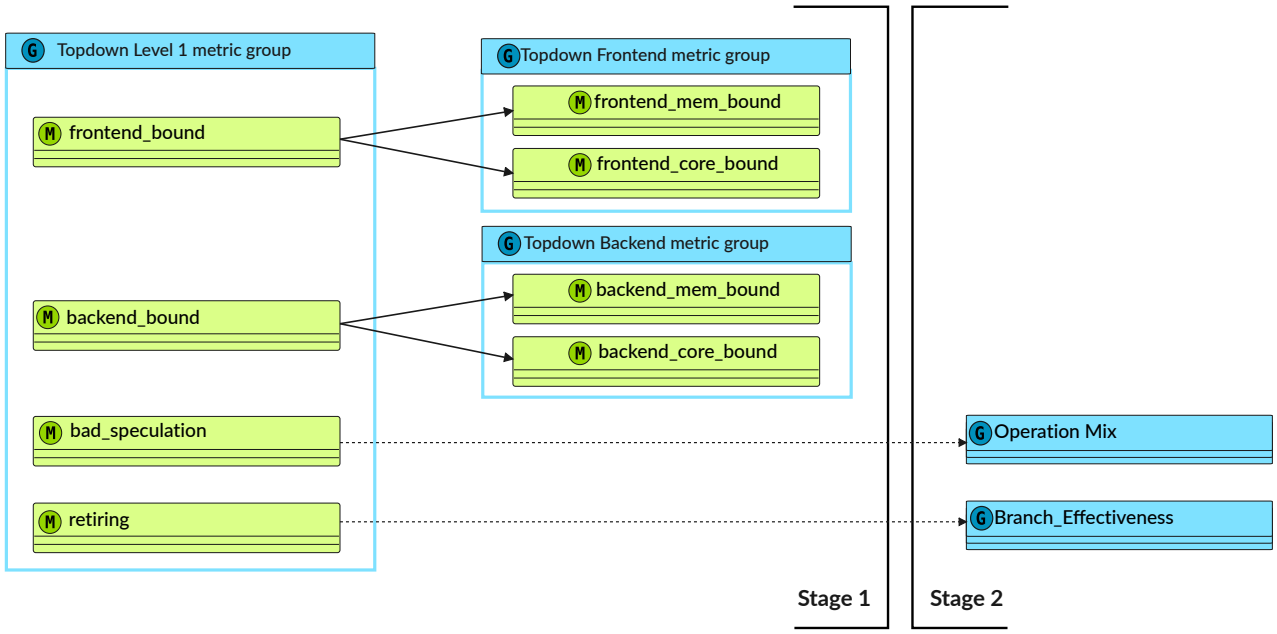
For more information about the metrics in this group and the associated formulas and events, see Topdown_CME.

3.1.1 Topdown Level 1 metric group

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

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

Figure 3-3: C1-Nano Metric Group: Topdown_L1



Topdown Level 1 metric group has four primary metrics Frontend Bound metric, Backend Bound metric, Bad Speculation metric, and Retiring metric.

Frontend Bound metric

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

To analyze the data further, use the next step Stage 1 Topdown Frontend metric group.

Backend Bound metric

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

To analyze the data further, use the next step Stage 1 Topdown Backend metric group.

Bad Speculation metric

The bad_speculation metric is the percentage of total slots that executed operations and did not retire due to a pipeline flush.

This indicates cycles that were utilized but inefficiently. This metric measures pipeline inefficiency caused by flushes in the pipeline, due to branch mispredictions or machine clears. For workloads that demonstrate high bad speculation rates, the next step is to understand the branch performance in the workload.

Some key reasons for pipeline inefficiency are the same reasons that have been identified in the `frontend_core_flush_bound` metric. This metric is part of the `frontend_core_bound` metric within the Topdown Frontend metric group section test.

To analyze the data further, use the following next steps where applicable:

- Stage 1 frontend_core_flush_bound metric
- Stage 2 Branch Effectiveness metric group

Retiring metric

The retiring metric is the percentage of total slots that retired operations, which indicates cycles that were utilized efficiently. This metric covers the total slots that were utilized efficiently by the processor.

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

To analyze the data further, use the following next steps where applicable:

- Stage 2 Operation Mix metric group

Topdown Level 1 implementation criteria

This metric group has criteria that apply to the implementation of the methodology.

- The sum of the metrics in Topdown_L1 equals 100% of the total execution cycles in the core implementation.
- A frontend stall is counted when there are no micro-ops to dispatch.
- A backend stall is counted when there are micro-ops in the dispatch unit, but they cannot dispatch to the backend due to backend resource constraints.

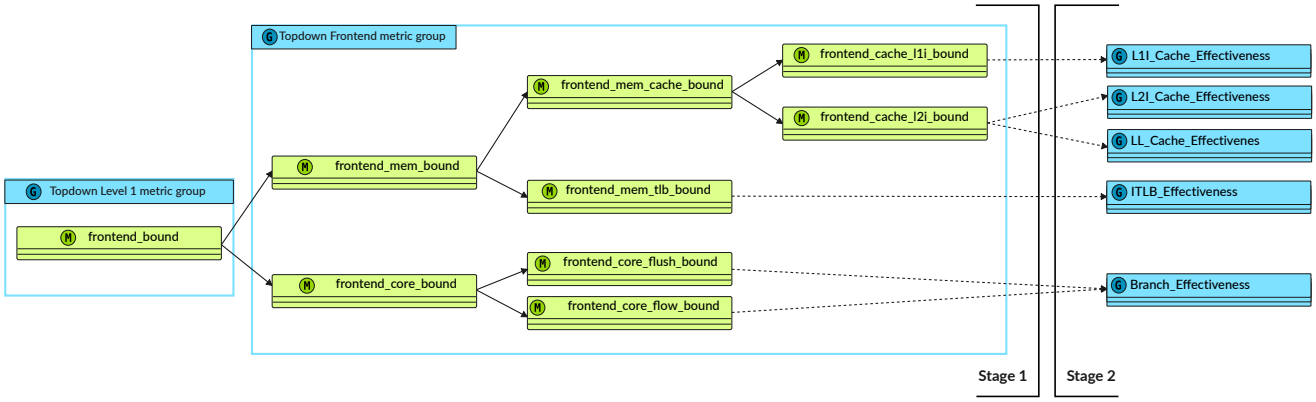
- In a cycle where there are no micro-ops to dispatch and no micro-ops dispatched to the backend, the CPU counts a frontend stall.
- The definition of the backend stall is linked to the slot that does not dispatch although there are micro-ops available in the dispatch unit.

3.1.2 Topdown Frontend metric group

The Topdown_Frontend metric group contains a set of metrics to analyze a frontend bound workload. The metric group divides frontend bound into memory bound and core bound.

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

Figure 3-4: C1-Nano Metric Group: Topdown_Frontend



The Topdown Frontend metric group has two primary metrics, Frontend Memory Bound metric and Frontend Core Bound metric.

Frontend Memory Bound metric

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

This metric breaks down into the following metrics:

- frontend_mem_cache_bound
  - frontend_cache_l1i_bound
  - frontend_cache_l2i_bound
- frontend_mem_tlb_bound

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

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

The following implementation criteria apply to these metrics:

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

To analyze the data further, use the following next steps where applicable:

- For `frontend_cache_l1i_bound`, use Stage 2 Level 1 Instruction Cache Effectiveness metric group.
- For `frontend_cache_l2i_bound`, use Stage 2 Level 2 Instruction Unified Cache Effectiveness metric group and Stage 2 Last Level Cache Effectiveness metric group.
- For `frontend_mem_tlb_bound`, use Stage 2 Instruction TLB Effectiveness metric group.

Frontend Core Bound metric

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

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

- Flush: Refers to the frontend resteer that cause PC updates, which can either be:
  - A flush that is caused by mispredictions
  - Other machine clears due to microarchitectural reasons such as exceptions and memory hazards. However, these cases are extremely rare.
- Flow: Refers to the frontend structural flow stall that is caused in the decode unit fetches because of a branch prediction unit that does not provide predictions on time.

This metric breaks down into the following metrics:

- frontend_core_flush_bound
- frontend_core_flow_bound

To analyze the data further, use the next step Stage 2 Branch Effectiveness metric group.

Topdown Frontend implementation criteria

This metric group has criteria that apply to the implementation of the methodology.

- The implementation of the Frontend Core Bound metric in C1-Nano is the difference between the Frontend Bound and Frontend Memory Bound metrics. This means that the cycle was stalled in the frontend and the stall was not caused by the frontend memory units. This derivation is mathematically correct but misses a direct count of the Frontend Core Bound metric from an implementation standpoint.

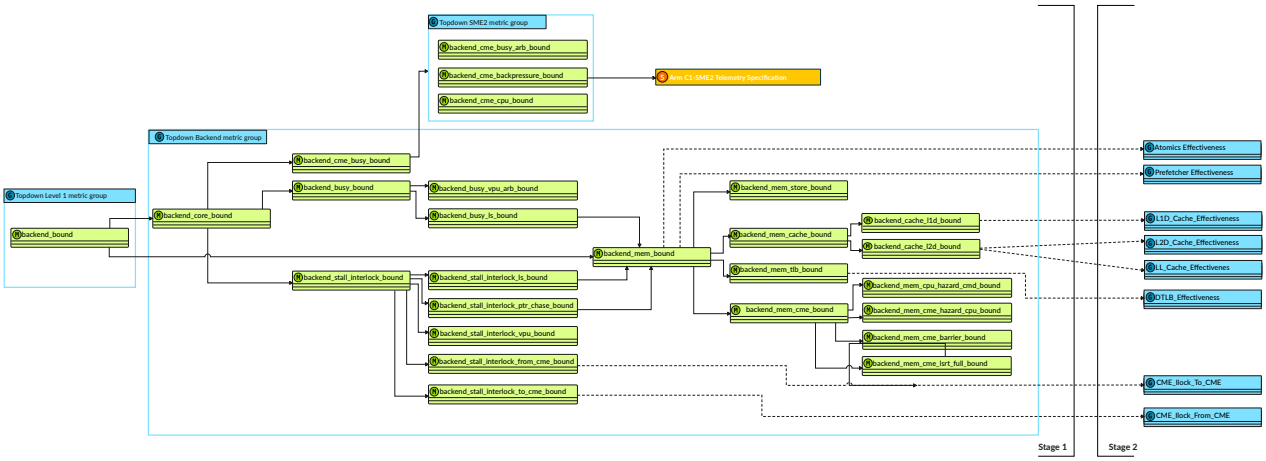
- Due to the implementation, the sum of the Frontend Core Bound and Frontend Memory Bound metrics is equal to 100% of the Frontend Bound metric.

3.1.3 Topdown Backend metric group

The Topdown_Backend metric group contains a set of metrics to analyze a backend bound workload. The metric group divides backend bound into memory bound and core bound.

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

Figure 3-5: C1-Nano Metric Group: Topdown_Backend



Topdown Backend metric group has two primary metrics, Backend Memory Bound metric and Backend Core Bound metric.

Backend Memory Bound metric

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

`backend_mem_bound` counts the stall cycles in which no operation can be sent to the dispatch unit and a memory operation is being executed. Additional stalls are introduced due to SME2 unit and CPU sharing the CPU memory subsystem. The SME2 unit and CPU are synchronized through the Load Store Registers (LSRT) block, causing both execution units to have dependencies on data accesses. New stalls can be analyzed using metrics starting with `backend_mem_cme_bound`.

This metric breaks down into the following metrics:

- backend_mem_cache_bound
  - backend_cache_l1d_bound
  - backend_cache_l2d_bound

- backend_mem_tlb_bound
- backend_mem_store_bound
- backend_mem_cme_bound
  - backend_mem_cme_barrier_bound
  - backend_mem_cme_lsrt_full_bound
  - backend_mem_cpu_hazard_cme_bound
  - backend_mem_cme_hazard_cpu_bound

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

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

The following implementation criteria apply to these metrics:

- The Backend Memory Cache Bound metric is implemented as the number of cycles of either the Backend Memory L1D Bound metric or the Backend Memory L1D Bound metric.
- The Backend Memory L1D Bound metric counts when waiting on an L1 miss but not waiting on an L2 miss.
- The Backend Memory Store Bound metric also counts backend memory bound stall cycles while a store cannot be retired because of a structural hazard against the merging store buffer.

To analyze the data further, use the following steps where applicable:

- For `backend_cache_l1d_bound`, use Stage 2 Level 1 Data Cache Effectiveness metric group.
- For `backend_cache_l2d_bound`, use Stage 2 Level 2 Data Unified Cache Effectiveness metric group and Stage 2 Last Level Cache Effectiveness metric group.
- For `backend_mem_tlb_bound`, use Stage 2 Data TLB Effectiveness metric group.
- For `backend_mem_bound`, use Stage 2 Atomics Effectiveness metric group and Stage 2 Prefetcher Effectiveness metric group.

There is no further analysis for:

- `backend_mem_store_bound`
- `backend_mem_cme_barrier_bound`
- `backend_mem_cpu_hazard_cme_bound`
- `backend_mem_cme_hazard_cme_bound`
- `backend_mem_cme_lsrt_full_bound`

Backend Core Bound metric

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

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

- Backend busy: Refers to execution stalls caused by execution units stalls, primarily load/store (LS), VPU, or a combination. If the rate of ls-bound backend busy stalls is high, as indicated by the next-level `backend_busy_ls_bound` metric, Arm recommends analysis of the Backend Memory Bound metric and its subsequent breakdown. The Load Store Unit's outstanding request capacity is limited. It can be significantly impacted by cache or TLB misses that cause subsequent backpressure in the form of execution unit stalls.
- Interlock stalls: Refers to dispatch and issue stalls caused by operand dependencies, where the operand is not yet ready. Micro-operation execution is strictly in order. Therefore, operand dependencies where data is not sufficiently produced on time can be a significant contributor to backend stalls.

  A high rate of interlock stalls can indicate sub-optimal instruction scheduling for an in-order core. Interlock stalls can be further broken down into the following categories. These categories are not mutually exclusive, because a bundle of operations can depend on different source operands.

  - VPU-bound interlock: Identifies the shared vector unit as the producer of at least one operand that is not ready yet. This operand prevents a bundle of operations from being issued.
  - LS-bound interlock: Identifies a load or store as the producer of at least one operand that is not ready yet. This operand prevents a bundle of operations from being issued. Refer to the Backend Memory Bound metric and subsequent breakdown to check whether that the interlock is primarily affected by:
    - Memory resources such as caches or TLBs
    - Load-to-use latency

      If the Backend Memory Bound metric is not significant compared to the Backend Core Bound metric, the compiler's instruction scheduling can be a major contributing factor.
    - LS-bound pointer chase interlock: Identifies the specific subset of LS-bound interlock where the consumer is the address operand of a load/store instruction. Pointer chase code without sufficient distance in the number of instructions between the producing load and the dependent load/store can be a major contributing factor to the slowdown in an in-order core.

This metric breaks down into the following metrics:

- backend_cme_busy_bound
- backend_busy_bound
- backend_stall_interlock_bound

To analyze the data further, use the following steps where applicable:

- For `backend_core_cme_bound`, use Topdown SME2 metric group.
- For `backend_busy_ls_bound`, use `backend_mem_bound`.
- For `backend_stall_interlock_ls_bound`, use `backend_mem_bound`.
- For `backend_stall_interlock_ptr_chase_bound`, use `backend_mem_bound`.
- For `backend_stall_interlock_from_cme_bound`, use Stage 2 [Interlock to SME2 metric group].
- For `backend_stall_interlock_to_cme_bound`, use Stage 2 [Interlock from SME2 metric group].
- There is no further analysis for `backend_stall_interlock_vpu_bound`.

Topdown Backend implementation criteria

This metric group has criteria that apply to the implementation of the methodology.

- The implementation of the Backend Core Bound metric in C1-Nano is the difference between the Backend Bound and Backend Memory Bound metrics. This means that the cycle was stalled in the backend and the stall was not caused by the backend memory units. This derivation is mathematically correct but misses a direct count of the Backend Core Bound metric from an implementation standpoint.
- Due to the implementation, the sum of the Backend Core Bound and Backend Memory Bound metrics is equal to 100% of the Backend Bound metric.

3.1.4 Topdown SME2 metric group

The Topdown SME2 metric group contains a set of metrics to analyze an SME2 bound workload.

This metric breaks down into the following metrics:

- backend_cme_backpressure_bound
- backend_cme_busy_arb_bound
- backend_cme_cpu_bound

To analyze the data further, use the following next steps where applicable:

- For `backend_cme_backpressure_bound`, analyze the bottlenecks that are caused by the resources that the SME2 unit manages, as described in the *Arm® C1-Scalable Matrix Extension 2 Telemetry Specification*.
- There is no further analysis for:
  - `backend_cme_busy_arb_bound`
  - `backend_cme_cpu_bound`

Topdown SME2 implementation criteria

This metric group has criteria that apply to the implementation of the methodology for the SME2 unit.

For more information, see the *Arm® C1-Scalable Matrix Extension 2 Telemetry Specification*.

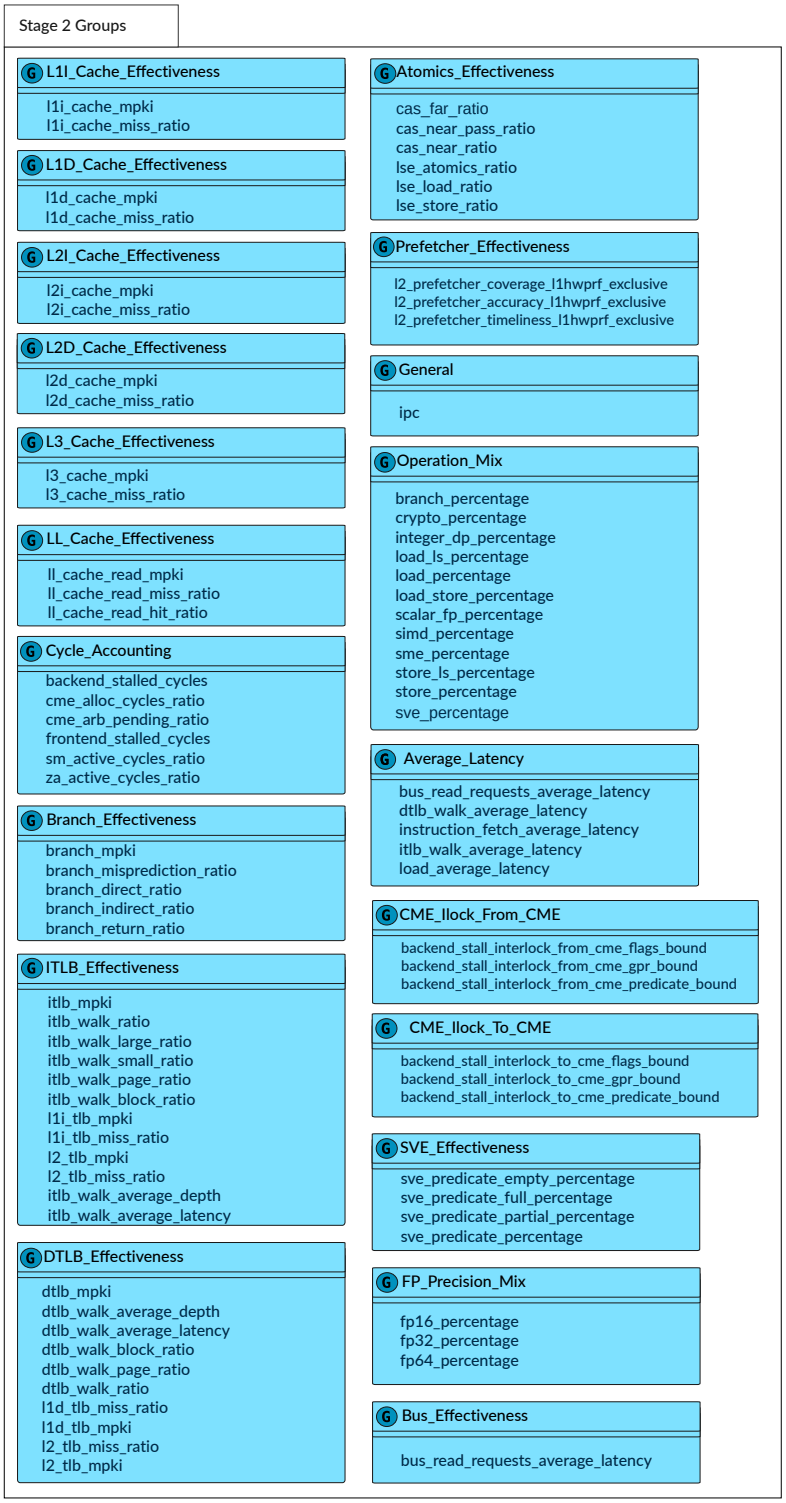
3.2 Stage 2: Microarchitecture exploration

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

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

The following figure shows the Stage 2 metric groups for C1-Nano.

Figure 3-6: C1-Nano Topdown methodology Stage 2 overview



In C1-Nano there are two metric groups based on industry standard metrics, Misses Per Kilo Instructions (MPKI) and Miss Ratios, plus several other Stage 2 metric groups.

Level 1 Data Cache Effectiveness metric group

The L1D_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the L1 Data Cache on the processor.

The metrics in this metric group include cache misses per thousand instructions and miss to access ratio.

Level 1 Instruction Cache Effectiveness metric group

The L1I_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the L1 Instruction Cache on the processor.

The metrics in this metric group include cache misses per thousand instructions and miss to access ratio.

Level 2 Data Unified Cache Effectiveness metric group

The L2D_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the L2 Unified Cache data accesses on the processor.

The metrics in this metric group include cache misses per thousand instructions and miss to access ratio.

Level 2 Instruction Unified Cache Effectiveness metric group

The L2I_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the L2 Unified Cache instruction accesses on the processor.

The metrics in this metric group include cache misses per thousand instructions and miss to access ratio.

Level 3 Cache Effectiveness metric group

The L3_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the L3 Unified Cache on this processor.

Last Level Cache Effectiveness metric group

The LL_Cache_Effectiveness metric group contains metrics to evaluate the effectiveness of the Last Level Cache on the processor.

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

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

The ll_cache_read_mpki and ll_cache_read_miss_ratio metrics can be used to analyze the last level read behavior.

Another useful metric to measure the SLC hit percentage for read traffic is ll_cache_read_hit_ratio.

Last level cache events do not have a write variant in C1-Nano because the SLC is only used as an eviction cache for the core. In addition, all the writes complete early at the interconnect when the transaction is acknowledged but always completed.

Data TLB Effectiveness and Instruction TLB Effectiveness metric groups

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

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

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

Note

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

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

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

- The itlb_mpki and dtlb_mpki metrics provide the rate of TLB Walks per kilo instructions for instruction and data accesses, respectively. These metrics help to evaluate and correlate the TLB efficiency with respect to the total number of instructions.
- The dtlb_walk_ratio metric provides the ratio of DTLB Walks to the overall TLB lookups made by the program. This metric is the same as the DTLB_WALK and MEM_ACCESS events because every MEM_ACCESS causes a L1D_TLB access.
- The itlb_walk_ratio metric provides a percentage of ITLB walks to the overall TLB lookups initiated from the instruction side.

Metrics are defined to provide insight into the depth of translation tables walks that utilize the ITLB_STEP and DTLB_STEP events. These counters count all memory accesses to both stage 1 and stage 2 translation tables for a translation table walk. Address translation mappings with an excessive number of steps per translation might impact performance.

Additionally, metrics are defined to provide ratios of walks that result in blocks versus pages which utilize ITLB_WALK_BLOCK, ITLB_WALK_PAGE,DTLB_WALK_BLOCK, and DTLB_WALK_PAGE events.


Note

The implementation for this CPU defines BLOCK/LARGE to be greater to or equal to to 2MB, and PAGE/SMALL to be less than 2MB.

Atomics Effectiveness metric group

The Atomics_Effectiveness metric group contains metrics to evaluate the effectiveness of atomics execution on the processor.

The metric group includes metrics for CAS atomics, as well as LSE atomics. The metric group provides metrics to analyze passing and failing atomic operations.

Average Latency metric group

The Average_Latency metric group contains metrics that provide average latencies of costly memory related operations by the processor that can take a variable number of cycles to execute.

The metrics in this metric group utilize PMU counters that aggregate the number of outstanding operations per cycle such as MEM_ACCESS_RD_PERCYC to derive the average latency.

Bus Effectiveness metric group

The Bus_Effectiveness metric group contains metrics to evaluate the effectiveness of bus transactions issued by this processor.

Branch Effectiveness metric group

The Branch_Effectiveness metric group contains metrics to evaluate the effectiveness of branch instruction execution on the processor.

Branch mispredictions are costly in a deeply pipelined CPU, causing pipeline flushes and wasted cycles. Although modern CPUs have optimized branch prediction units, there are many use cases that are branch heavy and hard to predict.

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

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

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

Branch History Table (BHT)
: Stores the history of conditional branches, taken or not.

Branch Target Buffer (BTB)
: Stores the target address for indirect branches.

Return Address Stack (RAS)
: Stores the function return branches.

C1-Nano supports the following three events that categorize the immediate, indirect, and return branches executed, respectively:

- BR_IMMED_SPEC
- BR_INDIRECT_SPEC
- BR_RETURN_SPEC

You can get a breakdown of the branch type to help investigate each subunit within the branch prediction unit.

Operation Mix metric group

The Operation_Mix metric group provides the distribution of micro-operation types executed for the program.

The C1-Nano microarchitecture has a variety of execution units that can process more than seven types of operations:

- Branch
- Single-cycle integers
- Multicycle integers
- Load Store Unit with address generation
- Advanced FP/SIMD operations
- Scalable Vector Extension (SVE)
- Scalable Matrix Extension (SME)

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

Note

Load/store instructions that behave as both a load and a store are reflected once in the load_store_percentage.

Prefetcher Effectiveness metric group

The Prefetcher_Effectiveness metric group contains metrics to evaluate the effectiveness of the L1 and L2 data prefetchers on the processor.

These metrics include:

- Coverage, a measure of eliminated prefetches
- Accuracy, a measure of useful prefetches
- Timeliness, a measure of appropriate timing of prefetches

SVE Effectiveness metric group

The SVE_Effectiveness metric group provides metrics to evaluate the effectiveness of predicated SVE instruction execution on this processor.

Floating Point Precision metric group

The FP_Precision_Mix metric group provides metrics to evaluate tmix of precision of floating point instruction execution on this processor.

Interlock from CME metric group

The CME_Ilock_From_CME metric group contains a set of metrics that measure the percentage of processor cycles stalled in backend of the processor. The cause is the CPU instruction being stalled by a data hazard from the SME2 unit.

Interlock to CME metric group

The CME_Ilock_To_CME metric group contains a set of metrics that measure the percentage of processor cycles stalled in backend of the processor. The cause is the oldest SME2 instruction being stalled by a data hazard.

Cycle Accounting metric group

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

General metric group

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

Misses Per Kilo Instructions metric group

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

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

Note

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

Miss Ratio metric group

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

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


Note

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

3.3 Core memory operations

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

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

C1-Nano also supports two additional events, MEM_ACCESS_RD and MEM_ACCESS_WR that can provide the read and write traffic breakdown respectively. These events are not the same as the LD_SPEC and ST_SPEC events because they count memory operations that are speculatively issued but not always executed.

3.4 System memory effectiveness

The System_Memory_Effectiveness metric group contains a set of metrics to analyze a system memory bound workload. They provide hierarchical data hits in system memory resources internal or external to the processor.

This metric group provides memory access metrics for DRAM, L3 Cache, last level Cache, and peer cluster cache.

For systems with multiple sockets or SoCs, C1-Nano supports the REMOTE_ACCESS event, which counts the memory transactions that were completed by a subordinate source from another chip.

4. C1-Nano Telemetry cheat-sheets and lookup tables

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

Cheat-sheets

Both metrics and events are listed by metric groups.

Lookup tables

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

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

4.1 Metrics cheat sheet for C1-Nano

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

C1-Nano specification provides the following types of metrics:

- Total implemented Common metrics: 119

Topdown Backend (22)	Interlock to SME2 (3)	Interlock from SME2 (3)
• backend_busy_bound • backend_busy_ls_bound • backend_busy_vpu_arb_bound • backend_cache_l1d_bound • backend_cache_l2d_bound • backend_core_bound • backend_core_cme_bound • backend_mem_bound • backend_mem_cache_bound • backend_mem_cme_barrier_bound • backend_mem_cme_bound • backend_mem_cme_hazard_cpu_bound • backend_mem_cme_lsrt_full_bound • backend_mem_cpu_hazard_cme_bound • backend_mem_store_bound • backend_mem_tlb_bound • backend_stall_interlock_bound • backend_stall_interlock_from_cme _bound • backend_stall_interlock_ls_bound • backend_stall_interlock_ptr_chase _bound • backend_stall_interlock_to_cme _bound • backend_stall_interlock_vpu_bound	• backend_stall_interlock_to_cme _flags_bound • backend_stall_interlock_to_cme_gpr _bound • backend_stall_interlock_to_cme _predicate_bound	• backend_stall_interlock_from_cme _flags_bound • backend_stall_interlock_from_cme _gpr_bound • backend_stall_interlock_from_cme _predicate_bound

Cycle Accounting (6)	Topdown SME2 (3)	Topdown Level 1 (4)
• backend_stalled_cycles • cme_alloc_cycles_ratio • cme_arb_pending_ratio • frontend_stalled_cycles • sm_active_cycles_ratio • za_active_cycles_ratio	• backend_cme_backpressure_bound • backend_cme_busy_arb_bound • backend_cme_cpu_bound	• backend_bound • bad_speculation • frontend_bound • retiring

Topdown Frontend (8)	General (1)	Misses Per Kilo Instructions (12)
• frontend_cache_l1i_bound • frontend_cache_l2i_bound • frontend_core_bound • frontend_core_flow_bound • frontend_core_flush_bound • frontend_mem_bound • frontend_mem_cache_bound • frontend_mem_tlb_bound	• ipc	• branch_mpki • dtlb_mpki • itlb_mpki • l1d_cache_mpki • l1d_tlb_mpki • l1i_cache_mpki • l1i_tlb_mpki • l2_tlb_mpki • l2d_cache_mpki • l2i_cache_mpki • l3_cache_mpki • ll_cache_read_mpki

Miss Ratio (12)	SVE Effectiveness (4)	Floating Point Precision (3)
• branch_misprediction_ratio • dtlb_walk_ratio • itlb_walk_ratio • l1d_cache_miss_ratio • l1d_tlb_miss_ratio • l1i_cache_miss_ratio • l1i_tlb_miss_ratio • l2_tlb_miss_ratio • l2d_cache_miss_ratio • l2i_cache_miss_ratio • l3_cache_miss_ratio • ll_cache_read_miss_ratio	• sve_predicate_empty_percentage • sve_predicate_full_percentage • sve_predicate_partial_percentage • sve_predicate_percentage	• fp16_percentage • fp32_percentage • fp64_percentage

Branch Effectiveness (5)	Instruction TLB Effectiveness (10)	Data TLB Effectiveness (10)
• branch_direct_ratio • branch_indirect_ratio • branch_misprediction_ratio • branch_mpki • branch_return_ratio	• itlb_mpki • itlb_walk_average_depth • itlb_walk_average_latency • itlb_walk_block_ratio • itlb_walk_page_ratio • itlb_walk_ratio • l1i_tlb_miss_ratio • l1i_tlb_mpki • l2_tlb_miss_ratio • l2_tlb_mpki	• dtlb_mpki • dtlb_walk_average_depth • dtlb_walk_average_latency • dtlb_walk_block_ratio • dtlb_walk_page_ratio • dtlb_walk_ratio • l1d_tlb_miss_ratio • l1d_tlb_mpki • l2_tlb_miss_ratio • l2_tlb_mpki

L1 Instruction Cache Effectiveness (2)	L1 Data Cache Effectiveness (2)	L2I Unified Cache Effectiveness (2)
• l1i_cache_miss_ratio • l1i_cache_mpki	• l1d_cache_miss_ratio • l1d_cache_mpki	• l2i_cache_miss_ratio • l2i_cache_mpki

L2D Unified Cache Effectiveness (2)	L3 Unified Cache Effectiveness (2)	Last Level Cache Effectiveness (3)
• l2d_cache_miss_ratio • l2d_cache_mpki	• l3_cache_miss_ratio • l3_cache_mpki	• ll_cache_read_hit_ratio • ll_cache_read_miss_ratio • ll_cache_read_mpki

Speculative Operation Mix (12)	Prefetcher Effectiveness (3)	Average Latency (5)
• branch_percentage • crypto_percentage • integer_dp_percentage • load_ls_percentage • load_percentage • load_store_percentage • scalar_fp_percentage • simd_percentage • sme_percentage • store_ls_percentage • store_percentage • sve_percentage	• l2_prefetcher_accuracy_l1hwprf_exclusive • l2_prefetcher_coverage_l1hwprf_exclusive • l2_prefetcher_timeliness_l1hwprf_exclusive	• bus_read_requests_average_latency • dtlb_walk_average_latency • instruction_fetch_average_latency • itlb_walk_average_latency • load_average_latency

Bus Effectiveness (1)	System Memory Effectiveness (2)	Atomics Effectiveness (6)
• bus_read_requests_average_latency	• system_l3_cache_hit_ratio • system_peer_cluster_cache_hit_ratio	• cas_far_ratio • cas_near_pass_ratio • cas_near_ratio • lse_atomics_ratio • lse_load_ratio • lse_store_ratio

4.2 PMU events cheat sheet for C1-Nano

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

C1-Nano specification provides the following types of PMU events:

- Total implemented Common events: 308
- Total Implemented Product ImpDef events: 89
- PMU Only events : 89
- ETE Only events : 2

Topdown Backend (22)	Interlock to SME2 (4)	Interlock from SME2 (4)
• IMP_STALL_BACKEND_BUSY_LS • IMP_STALL_BACKEND_BUSY_VPU_ARB • IMP_STALL_BACKEND_ILOCK_LS • IMP_STALL_BACKEND_ILOCK_LS_INTO_AGU • IMP_STALL_BACKEND_ILOCK_VPU • STALL_BACKEND • STALL_BACKEND_BUSY • STALL_BACKEND_BUSY_CME • STALL_BACKEND_CPUBOUND • STALL_BACKEND_ILOCK • STALL_BACKEND_ILOCK_FROM_CME • STALL_BACKEND_ILOCK_TO_CME • STALL_BACKEND_L1D • STALL_BACKEND_MEM • STALL_BACKEND_MEMBOUND • STALL_BACKEND_MEM_CME • STALL_BACKEND_MEM_CME_BARRIER • STALL_BACKEND_MEM_CME_HZ_ON_CPU • STALL_BACKEND_MEM_CME_LSRT_FULL • STALL_BACKEND_MEM_CPU_HZ_ON_CME • STALL_BACKEND_ST • STALL_BACKEND_TLB	• STALL_BACKEND_ILOCK_TO_CME • STALL_BACKEND_ILOCK_TO_CME_FLAGS • STALL_BACKEND_ILOCK_TO_CME_GPR • STALL_BACKEND_ILOCK_TO_CME_PRED	• STALL_BACKEND_ILOCK_FROM_CME • STALL_BACKEND_ILOCK_FROM_CME_FLAGS • STALL_BACKEND_ILOCK_FROM_CME_GPR • STALL_BACKEND_ILOCK_FROM_CME_PRED

Cycle Accounting (7)	Topdown SME2 (4)	Topdown Level 1 (7)
• CPU_CYCLES • CYCLES_ARB_PENDING_CME • CYCLES_CME_ALLOC • SM_ACTIVE_CYCLES • STALL_BACKEND • STALL_FRONTEND • ZA_ACTIVE	• STALL_BACKEND_BUSY_CME • STALL_BACKEND_BUSY_CMEBOUND • STALL_BACKEND_BUSY_CME_ARB • STALL_BACKEND_BUSY_CME_CPUBOUND	• CPU_CYCLES • OP_RETIRED • OP_SPEC • STALL_FRONTEND_FLUSH • STALL_SLOT • STALL_SLOT_BACKEND • STALL_SLOT_FRONTEND

Topdown Frontend (8)	General (2)	Misses Per Kilo Instructions (15)
• STALL_FRONTEND • STALL_FRONTEND_CPUBOUND • STALL_FRONTEND_FLOW • STALL_FRONTEND_FLUSH • STALL_FRONTEND_L1I • STALL_FRONTEND_MEM • STALL_FRONTEND_MEMBOUND • STALL_FRONTEND_TLB	• CPU_CYCLES • INST_RETIRED	• BR_MIS_PRED_RETIRED • DTLB_WALK • INST_RETIRED • ITLB_WALK • L1D_CACHE_REFILL_RD • L1D_CACHE_REFILL_WR • L1D_TLB_REFILL • L1I_CACHE_REFILL • L1I_TLB_REFILL • L2D_CACHE_REFILL_RD • L2D_CACHE_REFILL_WR • L2D_TLB_REFILL • L2I_CACHE_REFILL • L3D_CACHE_REFILL_RD • LL_CACHE_MISS_RD

Miss Ratio (24)	SVE Effectiveness (5)	Floating Point Precision (4)
• BR_MIS_PRED_RETIRED • BR_RETIRED • DTLB_WALK • ITLB_WALK • L1D_CACHE_REFILL_RD • L1D_CACHE_REFILL_WR • L1D_CACHE_RW • L1D_TLB • L1D_TLB_REFILL • L1I_CACHE • L1I_CACHE_REFILL • L1I_TLB • L1I_TLB_REFILL • L2D_CACHE_REFILL_RD • L2D_CACHE_REFILL_WR • L2D_CACHE_RW • L2D_TLB • L2D_TLB_REFILL • L2I_CACHE • L2I_CACHE_REFILL • L3D_CACHE_RD • L3D_CACHE_REFILL_RD • LL_CACHE_MISS_RD • LL_CACHE_RD	• INST_SPEC • SVE_PRED_EMPTY_SPEC • SVE_PRED_FULL_SPEC • SVE_PRED_PARTIAL_SPEC • SVE_PRED_SPEC	• FP_DP_SPEC • FP_HP_SPEC • FP_SP_SPEC • INST_SPEC

Branch Effectiveness (6)	Instruction TLB Effectiveness (10)	Data TLB Effectiveness (10)
• BR_IMMED_RETIRED • BR_IND_RETIRED • BR_MIS_PRED_RETIRED • BR_RETIRED • BR_RETURN_RETIRED • INST_RETIRED	• INST_RETIRED • ITLB_STEP • ITLB_WALK • ITLB_WALK_BLOCK • ITLB_WALK_PAGE • ITLB_WALK_PERCYC • L1I_TLB • L1I_TLB_REFILL • L2D_TLB • L2D_TLB_REFILL	• DTLB_STEP • DTLB_WALK • DTLB_WALK_BLOCK • DTLB_WALK_PAGE • DTLB_WALK_PERCYC • INST_RETIRED • L1D_TLB • L1D_TLB_REFILL • L2D_TLB • L2D_TLB_REFILL

L1 Instruction Cache Effectiveness (3)	L1 Data Cache Effectiveness (4)	L2I Unified Cache Effectiveness (3)
• INST_RETIRED • L1I_CACHE • L1I_CACHE_REFILL	• INST_RETIRED • L1D_CACHE_REFILL_RD • L1D_CACHE_REFILL_WR • L1D_CACHE_RW	• INST_RETIRED • L2I_CACHE • L2I_CACHE_REFILL

L2D Unified Cache Effectiveness (4)	L3 Unified Cache Effectiveness (3)	Last Level Cache Effectiveness (3)
• INST_RETIRED • L2D_CACHE_REFILL_RD • L2D_CACHE_REFILL_WR • L2D_CACHE_RW	• INST_RETIRED • L3D_CACHE_RD • L3D_CACHE_REFILL_RD	• INST_RETIRED • LL_CACHE_MISS_RD • LL_CACHE_RD

Speculative Operation Mix (12)	Prefetcher Effectiveness (5)	Average Latency (10)
• ASE_SPEC • BR_IMMED_SPEC • BR_INDIRECT_SPEC • CRYPTO_SPEC • DP_SPEC • INST_SPEC • LDST_SPEC • LD_SPEC • SME_INST_SPEC • ST_SPEC • SVE_SPEC • VFP_SPEC	• L2D_CACHE_HIT_RW_FHWPRF • L2D_CACHE_REFILL_HWPRF • L2D_CACHE_REFILL_RD • L2D_CACHE_REFILL_WR • L2D_LFB_HIT_RW_FHWPRF	• BUS_REQ_RD • BUS_REQ_RD_PERCYC • DTLB_WALK • DTLB_WALK_PERCYC • INST_FETCH • INST_FETCH_PERCYC • ITLB_WALK • ITLB_WALK_PERCYC • MEM_ACCESS • MEM_ACCESS_RD_PERCYC

Bus Effectiveness (2)	System Memory Effectiveness (6)	Atomics Effectiveness (7)
• BUS_REQ_RD • BUS_REQ_RD_PERCYC	• DSNP_HIT • ISNP_HIT_RD • L2D_CACHE_REFILL • L2D_CACHE_REFILL_RD • L2I_CACHE_REFILL • L3D_CACHE_HIT_RD	• CAS_NEAR_PASS • CAS_NEAR_SPEC • CAS_SPEC • LDST_SPEC • LSE_LDST_SPEC • LSE_LD_SPEC • LSE_ST_SPEC

4.3 Metrics lookup table for C1-Nano

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

Table 4-19: Metrics listed by name, with related events and metric groups

Metric Name	Formula from Events	Metric Groups
backend_bound	STALL_SLOT_BACKEND / (3 * CPU_CYCLES) * 100	• Topdown_L1
backend_busy_bound	STALL_BACKEND_BUSY / STALL_BACKEND * 100	• Topdown_Backend
backend_busy_ls_bound	IMP_STALL_BACKEND_BUSY_LS / STALL_BACKEND * 100	• Topdown_Backend
backend_busy_vpu_arb_bound	IMP_STALL_BACKEND_BUSY _VPU_ARB / STALL_BACKEND * 100	• Topdown_Backend
backend_cache_l1d_bound	STALL_BACKEND_L1D / (STALL_BACKEND_L1D + STALL_BACKEND_MEM) * 100	• Topdown_Backend
backend_cache_l2d_bound	STALL_BACKEND_MEM / (STALL_BACKEND_L1D + STALL_BACKEND_MEM) * 100	• Topdown_Backend
backend_cme_backpressure_bound	STALL_BACKEND_BUSY _CMEBOUND / STALL_BACKEND_BUSY_CME * 100	• Topdown_CME
backend_cme_busy_arb_bound	STALL_BACKEND_BUSY_CME _ARB / STALL_BACKEND_BUSY_CME * 100	• Topdown_CME
backend_cme_cpu_bound	STALL_BACKEND_BUSY_CME _CPUBOUND / STALL_BACKEND_BUSY_CME * 100	• Topdown_CME
backend_core_bound	STALL_BACKEND_CPUBOUND / STALL_BACKEND * 100	• Topdown_Backend
backend_core_cme_bound	STALL_BACKEND_BUSY_CME / STALL_BACKEND_CPUBOUND * 100	• Topdown_Backend
backend_mem_bound	STALL_BACKEND_MEMBOUND / STALL_BACKEND * 100	• Topdown_Backend

Metric Name	Formula from Events	Metric Groups
backend_mem_cache_bound	(STALL_BACKEND_L1D + STALL_BACKEND_MEM) / STALL_BACKEND_MEMBOUND * 100	• Topdown_Backend
backend_mem_cme_barrier_bound	STALL_BACKEND_MEM_CME _BARRIER / STALL_BACKEND_MEM_CME * 100	• Topdown_Backend
backend_mem_cme_bound	STALL_BACKEND_MEM_CME / STALL_BACKEND_MEMBOUND * 100	• Topdown_Backend
backend_mem_cme_hazard_cpu_bound	STALL_BACKEND_MEM_CME_HZ _ON_CPU / STALL_BACKEND_MEM_CME * 100	• Topdown_Backend
backend_mem_cme_lsrt_full_bound	STALL_BACKEND_MEM_CME _LSRT_FULL / STALL_BACKEND_MEM_CME * 100	• Topdown_Backend
backend_mem_cpu_hazard_cme_bound	STALL_BACKEND_MEM_CPU_HZ _ON_CME / STALL_BACKEND_MEM_CME * 100	• Topdown_Backend
backend_mem_store_bound	STALL_BACKEND_ST / STALL_BACKEND_MEMBOUND * 100	• Topdown_Backend
backend_mem_tlb_bound	STALL_BACKEND_TLB / STALL_BACKEND_MEMBOUND * 100	• Topdown_Backend
backend_stall_interlock_bound	STALL_BACKEND_ILOCK / STALL_BACKEND * 100	• Topdown_Backend
backend_stall_interlock_from_cme_bound	STALL_BACKEND_ILOCK_FROM _CME / STALL_BACKEND_ILOCK * 100	• Topdown_Backend
backend_stall_interlock_from_cme_flags _bound	STALL_BACKEND_ILOCK_FROM _CME_FLAGS / STALL_BACKEND_ILOCK_FROM _CME * 100	• CME_Ilock_From_CME
backend_stall_interlock_from_cme_gpr_bound	STALL_BACKEND_ILOCK_FROM _CME_GPR / STALL_BACKEND_ILOCK_FROM _CME * 100	• CME_Ilock_From_CME
backend_stall_interlock_from_cme_predicate _bound	STALL_BACKEND_ILOCK_FROM _CME_PRED / STALL_BACKEND_ILOCK_FROM _CME * 100	• CME_Ilock_From_CME
backend_stall_interlock_ls_bound	IMP_STALL_BACKEND_ILOCK _LS / STALL_BACKEND * 100	• Topdown_Backend
backend_stall_interlock_ptr_chase_bound	IMP_STALL_BACKEND_ILOCK _LS_INTO_AGU / STALL_BACKEND * 100	• Topdown_Backend
backend_stall_interlock_to_cme_bound	STALL_BACKEND_ILOCK_TO _CME / STALL_BACKEND_ILOCK * 100	• Topdown_Backend
backend_stall_interlock_to_cme_flags_bound	STALL_BACKEND_ILOCK_TO _CME_FLAGS / STALL_BACKEND_ILOCK_TO _CME * 100	• CME_Ilock_To_CME
backend_stall_interlock_to_cme_gpr_bound	STALL_BACKEND_ILOCK_TO _CME_GPR / STALL_BACKEND_ILOCK_TO _CME * 100	• CME_Ilock_To_CME
backend_stall_interlock_to_cme_predicate _bound	STALL_BACKEND_ILOCK_TO _CME_PRED / STALL_BACKEND_ILOCK_TO _CME * 100	• CME_Ilock_To_CME
backend_stall_interlock_vpu_bound	IMP_STALL_BACKEND_ILOCK _VPU / STALL_BACKEND * 100	• Topdown_Backend

Metric Name	Formula from Events	Metric Groups
backend_stalled_cycles	STALL_BACKEND / CPU_CYCLES * 100	• Cycle_Accounting
bad_speculation	(1 - STALL_SLOT / (3 * CPU_CYCLES)) * (1 - OP_RETIRED / OP_SPEC) * 100 + STALL_FRONTEND_FLUSH / CPU_CYCLES * 100	• Topdown_L1
branch_direct_ratio	BR_IMMED_RETIRED / BR_RETIRED	• Branch_Effectiveness
branch_indirect_ratio	BR_IND_RETIRED / BR_RETIRED	• Branch_Effectiveness
• branch_misprediction_ratio in Branch_Effectiveness • branch_misprediction_ratio in Miss_Ratio	BR_MIS_PRED_RETIRED / BR_RETIRED	• Branch_Effectiveness • Miss_Ratio
• branch_mpki in Branch_Effectiveness • branch_mpki in MPKI	BR_MIS_PRED_RETIRED / INST_RETIRED * 1000	• Branch_Effectiveness • MPKI
branch_percentage	(BR_IMMED_SPEC + BR_INDIRECT_SPEC) / INST_SPEC * 100	• Operation_Mix
branch_return_ratio	BR_RETURN_RETIRED / BR_RETIRED	• Branch_Effectiveness
• bus_read_requests_average_latency in Average_Latency • bus_read_requests_average_latency in Bus_Effectiveness	BUS_REQ_RD_PERCYC / BUS_REQ_RD	• Average_Latency • Bus_Effectiveness
cas_far_ratio	1 - CAS_NEAR_SPEC / CAS_SPEC	• Atomics_Effectiveness
cas_near_pass_ratio	CAS_NEAR_PASS / CAS_NEAR_SPEC	• Atomics_Effectiveness
cas_near_ratio	CAS_NEAR_SPEC / CAS_SPEC	• Atomics_Effectiveness
cme_alloc_cycles_ratio	CYCLES_CME_ALLOC / CPU_CYCLES * 100	• Cycle_Accounting
cme_arb_pending_ratio	CYCLES_ARB_PENDING_CME / CPU_CYCLES * 100	• Cycle_Accounting
crypto_percentage	CRYPTO_SPEC / INST_SPEC * 100	• Operation_Mix
• dtlb_mpki in DTLB_Effectiveness • dtlb_mpki in MPKI	DTLB_WALK / INST_RETIRED * 1000	• DTLB_Effectiveness • MPKI
dtlb_walk_average_depth	DTLB_STEP / DTLB_WALK	• DTLB_Effectiveness
• dtlb_walk_average_latency in Average_Latency • dtlb_walk_average_latency in DTLB_Effectiveness	DTLB_WALK_PERCYC / DTLB_WALK	• Average_Latency • DTLB_Effectiveness
dtlb_walk_block_ratio	DTLB_WALK_BLOCK / L1D_TLB	• DTLB_Effectiveness
dtlb_walk_page_ratio	DTLB_WALK_PAGE / L1D_TLB	• DTLB_Effectiveness
• dtlb_walk_ratio in DTLB_Effectiveness • dtlb_walk_ratio in Miss_Ratio	DTLB_WALK / L1D_TLB	• DTLB_Effectiveness • Miss_Ratio
fp16_percentage	FP_HP_SPEC / INST_SPEC * 100	• FP_Precision_Mix
fp32_percentage	FP_SP_SPEC / INST_SPEC * 100	• FP_Precision_Mix
fp64_percentage	FP_DP_SPEC / INST_SPEC * 100	• FP_Precision_Mix
frontend_bound	(STALL_SLOT_FRONTEND / (3 * CPU_CYCLES) - STALL_FRONTEND_FLUSH / CPU_CYCLES) * 100	• Topdown_L1

Metric Name	Formula from Events	Metric Groups
frontend_cache_l1i_bound	STALL_FRONTEND_L1I / (STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) * 100	• Topdown_Frontend
frontend_cache_l2i_bound	STALL_FRONTEND_MEM / (STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) * 100	• Topdown_Frontend
frontend_core_bound	STALL_FRONTEND_CPUBOUND / STALL_FRONTEND * 100	• Topdown_Frontend
frontend_core_flow_bound	STALL_FRONTEND_FLOW / STALL_FRONTEND_CPUBOUND * 100	• Topdown_Frontend
frontend_core_flush_bound	STALL_FRONTEND_FLUSH / STALL_FRONTEND_CPUBOUND * 100	• Topdown_Frontend
frontend_mem_bound	STALL_FRONTEND_MEMBOUND / STALL_FRONTEND * 100	• Topdown_Frontend
frontend_mem_cache_bound	(STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) / STALL_FRONTEND_MEMBOUND * 100	• Topdown_Frontend
frontend_mem_tlb_bound	STALL_FRONTEND_TLB / STALL_FRONTEND_MEMBOUND * 100	• Topdown_Frontend
frontend_stalled_cycles	STALL_FRONTEND / CPU_CYCLES * 100	• Cycle_Accounting
instruction_fetch_average_latency	INST_FETCH_PERCYC / INST_FETCH	• Average_Latency
integer_dp_percentage	DP_SPEC / INST_SPEC * 100	• Operation_Mix
ipc	INST_RETIRED / CPU_CYCLES	• General
• itlb_mpki in ITLB_Effectiveness • itlb_mpki in MPKI	ITLB_WALK / INST_RETIRED * 1000	• ITLB_Effectiveness • MPKI
itlb_walk_average_depth	ITLB_STEP / ITLB_WALK	• ITLB_Effectiveness
• itlb_walk_average_latency in Average_Latency • itlb_walk_average_latency in ITLB_Effectiveness	ITLB_WALK_PERCYC / ITLB_WALK	• Average_Latency • ITLB_Effectiveness
itlb_walk_block_ratio	ITLB_WALK_BLOCK / L1I_TLB	• ITLB_Effectiveness
itlb_walk_page_ratio	ITLB_WALK_PAGE / L1I_TLB	• ITLB_Effectiveness
• itlb_walk_ratio in ITLB_Effectiveness • itlb_walk_ratio in Miss_Ratio	ITLB_WALK / L1I_TLB	• ITLB_Effectiveness • Miss_Ratio
• l1d_cache_miss_ratio in L1D_Cache_Effectiveness • l1d_cache_miss_ratio in Miss_Ratio	(L1D_CACHE_REFILL_RD + L1D_CACHE_REFILL_WR) / L1D_CACHE_RW	• L1D_Cache_Effectiveness • Miss_Ratio
• l1d_cache_mpki in L1D_Cache_Effectiveness • l1d_cache_mpki in MPKI	(L1D_CACHE_REFILL_RD + L1D_CACHE_REFILL_WR) / INST_RETIRED * 1000	• L1D_Cache_Effectiveness • MPKI
• l1d_tlb_miss_ratio in DTLB_Effectiveness • l1d_tlb_miss_ratio in Miss_Ratio	L1D_TLB_REFILL / L1D_TLB	• DTLB_Effectiveness • Miss_Ratio
• l1d_tlb_mpki in DTLB_Effectiveness • l1d_tlb_mpki in MPKI	L1D_TLB_REFILL / INST_RETIRED * 1000	• DTLB_Effectiveness • MPKI

Metric Name	Formula from Events	Metric Groups
• l1i_cache_miss_ratio in L1I_Cache_Effectiveness • l1i_cache_miss_ratio in Miss_Ratio	L1I_CACHE_REFILL / L1I_CACHE	• L1I_Cache_Effectiveness • Miss_Ratio
• l1i_cache_mpki in L1I_Cache_Effectiveness • l1i_cache_mpki in MPKI	L1I_CACHE_REFILL / INST_RETIRED * 1000	• L1I_Cache_Effectiveness • MPKI
• l1i_tlb_miss_ratio in ITLB_Effectiveness • l1i_tlb_miss_ratio in Miss_Ratio	L1I_TLB_REFILL / L1I_TLB	• ITLB_Effectiveness • Miss_Ratio
• l1i_tlb_mpki in ITLB_Effectiveness • l1i_tlb_mpki in MPKI	L1I_TLB_REFILL / INST_RETIRED * 1000	• ITLB_Effectiveness • MPKI
l2_prefetcher_accuracy_l1hwprf_exclusive	(L2D_CACHE_HIT_RW_FHWPRF + L2D_LFB_HIT_RW_FHWPRF) / L2D_CACHE_REFILL_HWPRF	• Prefetcher_Effectiveness
l2_prefetcher_coverage_l1hwprf_exclusive	(L2D_CACHE_HIT_RW_FHWPRF + L2D_LFB_HIT_RW_FHWPRF) / (L2D_CACHE_HIT_RW_FHWPRF + L2D_LFB_HIT_RW_FHWPRF + L2D_CACHE_REFILL_RD + L2D_CACHE_REFILL_WR)	• Prefetcher_Effectiveness
l2_prefetcher_timeliness_l1hwprf_exclusive	L2D_CACHE_HIT_RW_FHWPRF / (L2D_CACHE_HIT_RW_FHWPRF + L2D_LFB_HIT_RW_FHWPRF)	• Prefetcher_Effectiveness
• l2_tlb_miss_ratio in DTLB_Effectiveness • l2_tlb_miss_ratio in ITLB_Effectiveness • l2_tlb_miss_ratio in Miss_Ratio	L2D_TLB_REFILL / L2D_TLB	• DTLB_Effectiveness • ITLB_Effectiveness • Miss_Ratio
• l2_tlb_mpki in DTLB_Effectiveness • l2_tlb_mpki in ITLB_Effectiveness • l2_tlb_mpki in MPKI	L2D_TLB_REFILL / INST_RETIRED * 1000	• DTLB_Effectiveness • ITLB_Effectiveness • MPKI
• l2d_cache_miss_ratio in L2D_Cache_Effectiveness • l2d_cache_miss_ratio in Miss_Ratio	(L2D_CACHE_REFILL_RD + L2D_CACHE_REFILL_WR) / L2D_CACHE_RW	• L2D_Cache_Effectiveness • Miss_Ratio
• l2d_cache_mpki in L2D_Cache_Effectiveness • l2d_cache_mpki in MPKI	(L2D_CACHE_REFILL_RD + L2D_CACHE_REFILL_WR) / INST_RETIRED * 1000	• L2D_Cache_Effectiveness • MPKI
• l2i_cache_miss_ratio in L2I_Cache_Effectiveness • l2i_cache_miss_ratio in Miss_Ratio	L2I_CACHE_REFILL / L2I_CACHE	• L2I_Cache_Effectiveness • Miss_Ratio
• l2i_cache_mpki in L2I_Cache_Effectiveness • l2i_cache_mpki in MPKI	L2I_CACHE_REFILL / INST_RETIRED * 1000	• L2I_Cache_Effectiveness • MPKI
• l3_cache_miss_ratio in L3_Cache_Effectiveness • l3_cache_miss_ratio in Miss_Ratio	L3D_CACHE_REFILL_RD / L3D_CACHE_RD	• L3_Cache_Effectiveness • Miss_Ratio
• l3_cache_mpki in L3_Cache_Effectiveness • l3_cache_mpki in MPKI	L3D_CACHE_REFILL_RD / INST_RETIRED * 1000	• L3_Cache_Effectiveness • MPKI

Metric Name	Formula from Events	Metric Groups
ll_cache_read_hit_ratio	(LL_CACHE_RD - LL_CACHE_MISS_RD) / LL_CACHE_RD	• LL_Cache_Effectiveness
• ll_cache_read_miss_ratio in LL_Cache_Effectiveness • ll_cache_read_miss_ratio in Miss_Ratio	LL_CACHE_MISS_RD / LL_CACHE_RD	• LL_Cache_Effectiveness • Miss_Ratio
• ll_cache_read_mpki in LL_Cache_Effectiveness • ll_cache_read_mpki in MPKI	LL_CACHE_MISS_RD / INST_RETIRED * 1000	• LL_Cache_Effectiveness • MPKI
load_average_latency	MEM_ACCESS_RD_PERCYC / MEM_ACCESS	• Average_Latency
load_ls_percentage	LD_SPEC / LDST_SPEC * 100	• Operation_Mix
load_percentage	LD_SPEC / INST_SPEC * 100	• Operation_Mix
load_store_percentage	LDST_SPEC / INST_SPEC * 100	• Operation_Mix
lse_atomics_ratio	LSE_LDST_SPEC / LDST_SPEC	• Atomics_Effectiveness
lse_load_ratio	LSE_LD_SPEC / LSE_LDST_SPEC	• Atomics_Effectiveness
lse_store_ratio	LSE_ST_SPEC / LSE_LDST_SPEC	• Atomics_Effectiveness
retiring	(1 - STALL_SLOT / (CPU_CYCLES * 3)) * (OP_RETIRED / OP_SPEC) * 100	• Topdown_L1
scalar_fp_percentage	VFP_SPEC / INST_SPEC * 100	• Operation_Mix
simd_percentage	ASE_SPEC / INST_SPEC * 100	• Operation_Mix
sm_active_cycles_ratio	SM_ACTIVE_CYCLES / CPU_CYCLES * 100	• Cycle_Accounting
sme_percentage	SME_INST_SPEC / INST_SPEC * 100	• Operation_Mix
store_ls_percentage	ST_SPEC / LDST_SPEC * 100	• Operation_Mix
store_percentage	ST_SPEC / INST_SPEC * 100	• Operation_Mix
sve_percentage	SVE_SPEC / INST_SPEC * 100	• Operation_Mix
sve_predicate_empty_percentage	SVE_PRED_EMPTY_SPEC / SVE_PRED_SPEC * 100	• SVE_Effectiveness
sve_predicate_full_percentage	SVE_PRED_FULL_SPEC / SVE_PRED_SPEC * 100	• SVE_Effectiveness
sve_predicate_partial_percentage	SVE_PRED_PARTIAL_SPEC / SVE_PRED_SPEC * 100	• SVE_Effectiveness
sve_predicate_percentage	SVE_PRED_SPEC / INST_SPEC * 100	• SVE_Effectiveness
system_l3_cache_hit_ratio	L3D_CACHE_HIT_RD / (L2D_CACHE_REFILL_RD + L2I_CACHE_REFILL)	• System_Memory_Effectiveness
system_peer_cluster_cache_hit_ratio	(DSNP_HIT + ISNP_HIT_RD) / (L2D_CACHE_REFILL + L2I_CACHE_REFILL)	• System_Memory_Effectiveness
za_active_cycles_ratio	ZA_ACTIVE / CPU_CYCLES * 100	• Cycle_Accounting

4.4 PMU events lookup table for C1-Nano

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: 835
- Total implemented Common events: 308
  - Common : Architectural-defined events: 263
  - Common : Implementation-defined events: 45
- Total Implemented Product ImpDef events: 89
- PMU Only Events : 89
- ETE Only Events : 2

Table 4-20: Events listed by Event Code, with related Metrics, Metric Groups, and Functional Groups

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0000, SW_INCR	-	-	• Retired
0x0001, L1I_CACHE_REFILL	• 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	• L1I_Cache _Effectiveness • MPKI • Miss_Ratio	• L1I_Cache
0x0002, L1I_TLB_REFILL	• 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	• ITLB_Effectiveness • MPKI • Miss_Ratio	• TLB
0x0003, L1D_CACHE_REFILL	-	-	• L1D_Cache
0x0004, L1D_CACHE	-	-	• L1D_Cache
0x0005, L1D_TLB_REFILL	• 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	• DTLB_Effectiveness • MPKI • Miss_Ratio	• TLB
0x0006, LD_RETIRED	-	-	• Retired
0x0007, ST_RETIRED	-	-	• Retired

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0008, INST_RETIRED	• ipc • branch_mpki in Branch_Effectiveness • branch_mpki in MPKI • itlb_mpki in ITLB_Effectiveness • itlb_mpki in MPKI • l1i_tlb_mpki in ITLB_Effectiveness • l1i_tlb_mpki in MPKI • dtlb_mpki in DTLB_Effectiveness • dtlb_mpki in MPKI • l1d_tlb_mpki in DTLB_Effectiveness • l1d_tlb_mpki in MPKI • l2_tlb_mpki in DTLB_Effectiveness • l2_tlb_mpki in ITLB_Effectiveness • l2_tlb_mpki in MPKI • l1i_cache_mpki in L1I_Cache _Effectiveness • l1i_cache_mpki in MPKI • l1d_cache_mpki in L1D_Cache _Effectiveness • l1d_cache_mpki in MPKI • l2i_cache_mpki in L2I_Cache _Effectiveness • l2i_cache_mpki in MPKI • l2d_cache_mpki in L2D_Cache _Effectiveness • l2d_cache_mpki in MPKI • l3_cache_mpki in L3_Cache _Effectiveness • l3_cache_mpki in MPKI • ll_cache_read_mpki in LL_Cache _Effectiveness • ll_cache_read_mpki in MPKI	• Branch_Effectiveness • DTLB_Effectiveness • General • ITLB_Effectiveness • L1D_Cache _Effectiveness • L1I_Cache _Effectiveness • L2D_Cache _Effectiveness • L2I_Cache _Effectiveness • L3_Cache _Effectiveness • LL_Cache _Effectiveness • MPKI	• Retired
0x0009, EXC_TAKEN	-	-	• Exception
0x000A, EXC_RETURN	-	-	• Exception
0x000B, CID_WRITE_RETIRED	-	-	• Retired
0x000C, PC_WRITE_RETIRED	-	-	• Retired
0x000D, BR_IMMED_RETIRED	• branch_direct_ratio	• Branch_Effectiveness	• Retired
0x000E, BR_RETURN_RETIRED	• branch_return_ratio	• Branch_Effectiveness	• Retired

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0010, BR_MIS_PRED	-	-	• Spec_Operation
0x0011, CPU_CYCLES	• frontend_stalled_cycles • backend_stalled_cycles • frontend_bound • backend_bound • retiring • bad_speculation • ipc • sm_active_cycles_ratio • za_active_cycles_ratio • cme_alloc_cycles_ratio • cme_arb_pending_ratio	• Cycle_Accounting • General • Topdown_L1	• General
0x0012, BR_PRED	-	-	• Spec_Operation
0x0013, MEM_ACCESS	• load_average_latency	• Average_Latency	• Memory
0x0014, L1I_CACHE	• l1i_cache_miss_ratio in L1I_Cache _Effectiveness • l1i_cache_miss_ratio in Miss_Ratio	• L1I_Cache _Effectiveness • Miss_Ratio	• L1I_Cache
0x0015, L1D_CACHE_WB	-	-	• L1D_Cache
0x0016, L2D_CACHE	-	-	• L2_Cache
0x0017, L2D_CACHE_REFILL	• system_peer_cluster_cache _hit_ratio	• System_Memory _Effectiveness	• L2_Cache
0x0018, L2D_CACHE_WB	-	-	• L2_Cache
0x0019, BUS_ACCESS	-	-	• Bus
0x001A, MEMORY_ERROR	-	-	• Memory
0x001B, INST_SPEC	• load_store_percentage • load_percentage • store_percentage • integer_dp_percentage • simd_percentage • scalar_fp_percentage • branch_percentage • crypto_percentage • sve_percentage • sve_predicate_percentage • fp16_percentage • fp32_percentage • fp64_percentage • sme_percentage	• FP_Precision_Mix • Operation_Mix • SVE_Effectiveness	• Spec_Operation
0x001C, TTBR_WRITE_RETIRED	-	-	• Retired
0x001D, BUS_CYCLES	-	-	• Bus

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x001E, CHAIN	-	-	• Chain
0x0020, L2D_CACHE_ALLOCATE	-	-	• L2_Cache
0x0021, BR_RETIRED	• branch_direct_ratio • branch_indirect_ratio • branch_return_ratio • branch_misprediction _ratio in Branch_Effectiveness • branch_misprediction _ratio in Miss_Ratio	• Branch_Effectiveness • Miss_Ratio	• Retired
0x0022, BR_MIS_PRED_RETIRED	• branch_mpki in Branch_Effectiveness • branch_mpki in MPKI • branch_misprediction _ratio in Branch_Effectiveness • branch_misprediction _ratio in Miss_Ratio	• Branch_Effectiveness • MPKI • Miss_Ratio	• Retired
0x0023, STALL_FRONTEND	• frontend_stalled_cycles • frontend_core_bound • frontend_mem_bound	• Cycle_Accounting • Topdown_Frontend	• Stall
0x0024, STALL_BACKEND	• backend_stalled_cycles • backend_core_bound • backend_mem_bound • backend_stall_interlock _bound • backend_stall_interlock _ls_bound • backend_stall_interlock _ptr_chase_bound • backend_stall_interlock _vpu_bound • backend_busy_bound • backend_busy_ls_bound • backend_busy_vpu_arb _bound	• Cycle_Accounting • Topdown_Backend	• Stall
0x0025, L1D_TLB	• dtlb_walk_ratio in DTLB_Effectiveness • dtlb_walk_ratio in Miss_Ratio • dtlb_walk_page_ratio • dtlb_walk_block_ratio • l1d_tlb_miss_ratio in DTLB_Effectiveness • l1d_tlb_miss_ratio in Miss_Ratio	• DTLB_Effectiveness • Miss_Ratio	• TLB

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0026, L1I_TLB	• itlb_walk_ratio in ITLB_Effectiveness • itlb_walk_ratio in Miss_Ratio • itlb_walk_page_ratio • itlb_walk_block_ratio • l1i_tlb_miss_ratio in ITLB_Effectiveness • l1i_tlb_miss_ratio in Miss_Ratio	• ITLB_Effectiveness • Miss_Ratio	• TLB
0x0027, L2I_CACHE	• l2i_cache_miss_ratio in L2I_Cache _Effectiveness • l2i_cache_miss_ratio in Miss_Ratio	• L2I_Cache _Effectiveness • Miss_Ratio	• L2_Cache
0x0028, L2I_CACHE_REFILL	• l2i_cache_mpki in L2I_Cache _Effectiveness • l2i_cache_mpki in MPKI • l2i_cache_miss_ratio in L2I_Cache _Effectiveness • l2i_cache_miss_ratio in Miss_Ratio • system_l3_cache_hit_ratio • system_peer_cluster_cache _hit_ratio	• L2I_Cache _Effectiveness • MPKI • Miss_Ratio • System_Memory _Effectiveness	• L2_Cache
0x002B, L3D_CACHE	-	-	• L3_Cache
0x002D, L2D_TLB_REFILL	• l2_tlb_mpki in DTLB_Effectiveness • l2_tlb_mpki in ITLB_Effectiveness • l2_tlb_mpki in MPKI • l2_tlb_miss_ratio in DTLB_Effectiveness • l2_tlb_miss_ratio in ITLB_Effectiveness • l2_tlb_miss_ratio in Miss_Ratio	• DTLB_Effectiveness • ITLB_Effectiveness • MPKI • Miss_Ratio	• TLB
0x002E, L2I_TLB_REFILL	-	-	• TLB
0x002F, L2D_TLB	• l2_tlb_miss_ratio in DTLB_Effectiveness • l2_tlb_miss_ratio in ITLB_Effectiveness • l2_tlb_miss_ratio in Miss_Ratio	• DTLB_Effectiveness • ITLB_Effectiveness • Miss_Ratio	• TLB
0x0030, L2I_TLB	-	-	• TLB
0x0031, REMOTE_ACCESS	-	-	• Memory
0x0032, LL_CACHE	-	-	• LL_Cache

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0034, DTLB_WALK	• dtlb_mpki in DTLB_Effectiveness • dtlb_mpki in MPKI • dtlb_walk_ratio in DTLB_Effectiveness • dtlb_walk_ratio in Miss_Ratio • dtlb_walk_average_depth • dtlb_walk_average_latency in Average_Latency • dtlb_walk_average_latency in DTLB_Effectiveness	• Average_Latency • DTLB_Effectiveness • MPKI • Miss_Ratio	• TLB
0x0035, ITLB_WALK	• itlb_mpki in ITLB_Effectiveness • itlb_mpki in MPKI • itlb_walk_ratio in ITLB_Effectiveness • itlb_walk_ratio in Miss_Ratio • itlb_walk_average_depth • itlb_walk_average_latency in Average_Latency • itlb_walk_average_latency in ITLB_Effectiveness	• Average_Latency • ITLB_Effectiveness • MPKI • Miss_Ratio	• TLB
0x0036, LL_CACHE_RD	• ll_cache_read_miss_ratio in LL_Cache _Effectiveness • ll_cache_read_miss_ratio in Miss_Ratio • ll_cache_read_hit_ratio	• LL_Cache _Effectiveness • Miss_Ratio	• LL_Cache
0x0037, LL_CACHE_MISS_RD	• 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	• LL_Cache _Effectiveness • MPKI • Miss_Ratio	• LL_Cache
0x0038, REMOTE_ACCESS_RD	-	-	• Memory
0x0039, L1D_CACHE_LMISS_RD	-	-	• L1D_Cache
0x003A, OP_RETIRED	• retiring • bad_speculation	• Topdown_L1	• Retired
0x003B, OP_SPEC	• retiring • bad_speculation	• Topdown_L1	• Spec_Operation
0x003C, STALL	-	-	• Stall
0x003D, STALL_SLOT_BACKEND	• backend_bound	• Topdown_L1	• Stall

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x003E, STALL_SLOT_FRONTEND	• frontend_bound	• Topdown_L1	• Stall
0x003F, STALL_SLOT	• retiring • bad_speculation	• Topdown_L1	• Stall
0x0040, L1D_CACHE_RD	-	-	• L1D_Cache
0x0041, L1D_CACHE_WR	-	-	• L1D_Cache
0x0042, L1D_CACHE_REFILL_RD	• 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	• L1D_Cache _Effectiveness • MPKI • Miss_Ratio	• L1D_Cache
0x0043, L1D_CACHE_REFILL_WR	• 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	• L1D_Cache _Effectiveness • MPKI • Miss_Ratio	• L1D_Cache
0x0044, L1D_CACHE_REFILL_INNER	-	-	• L1D_Cache
0x0045, L1D_CACHE_REFILL_OUTER	-	-	• L1D_Cache
0x0046, L1D_CACHE_WB_VICTIM	-	-	• L1D_Cache
0x0050, L2D_CACHE_RD	-	-	• L2_Cache
0x0051, L2D_CACHE_WR	-	-	• L2_Cache
0x0052, L2D_CACHE_REFILL_RD	• l2d_cache_mpki in L2D_Cache _Effectiveness • l2d_cache_mpki in MPKI • l2d_cache_miss_ratio in L2D_Cache _Effectiveness • l2d_cache_miss_ratio in Miss_Ratio • l2_prefetcher_coverage _l1hwprf_exclusive • system_l3_cache_hit_ratio	• L2D_Cache _Effectiveness • MPKI • Miss_Ratio • Prefetcher _Effectiveness • System_Memory _Effectiveness	• L2_Cache

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0053, L2D_CACHE_REFILL_WR	• l2d_cache_mpki in L2D_Cache _Effectiveness • l2d_cache_mpki in MPKI • l2d_cache_miss_ratio in L2D_Cache _Effectiveness • l2d_cache_miss_ratio in Miss_Ratio • l2_prefetcher_coverage _l1hwprf_exclusive	• L2D_Cache _Effectiveness • MPKI • Miss_Ratio • Prefetcher _Effectiveness	• L2_Cache
0x0056, L2D_CACHE_WB_VICTIM	-	-	• L2_Cache
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_ls_percentage • load_percentage	• Operation_Mix	• Spec_Operation
0x0071, ST_SPEC	• store_ls_percentage • store_percentage	• Operation_Mix	• Spec_Operation
0x0072, LDST_SPEC	• load_store_percentage • load_ls_percentage • store_ls_percentage • lse_atomics_ratio	• Atomics _Effectiveness • 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

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x007F, CSDB_SPEC	-	-	• Spec_Operation
0x0086, EXC_IRQ	-	-	• Exception
0x0087, EXC_FIQ	-	-	• Exception
0x0090, RC_LD_SPEC	-	-	• Spec_Operation
0x0091, RC_ST_SPEC	-	-	• Spec_Operation
0x00A0, L3D_CACHE_RD	• l3_cache_miss_ratio in L3_Cache _Effectiveness • l3_cache_miss_ratio in Miss_Ratio	• L3_Cache _Effectiveness • Miss_Ratio	• L3_Cache
0x00A1, L3D_CACHE_WR	-	-	• L3_Cache
0x00A2, L3D_CACHE_REFILL_RD	• l3_cache_mpki in L3_Cache _Effectiveness • l3_cache_mpki in MPKI • l3_cache_miss_ratio in L3_Cache _Effectiveness • l3_cache_miss_ratio in Miss_Ratio	• L3_Cache _Effectiveness • MPKI • Miss_Ratio	• L3_Cache
0x00C3, IMP_L2D_WS_MODE	-	-	-
0x00C4, IMP_L1D_WS_MODE_ENTRY	-	-	-
0x00C5, IMP_L1D_WS_MODE	-	-	-
0x00C7, IMP_L3D_WS_MODE	-	-	-
0x00C8, IMP_LL_WS_MODE	-	-	-
0x00C9, IMP_L1D_CACHE_WR_NO_ALLOC	-	-	-
0x00CB, IMP_L2D_WS_MODE_WR_COUNT	-	-	-
0x00CC, IMP_L3D_WS_MODE_WR_COUNT	-	-	-
0x00CD, IMP_LL_WS_MODE_WR_COUNT	-	-	-
0x00D0, IMP_L2D_WALK_TLB	-	-	-
0x00D1, IMP_L2D_WALK_TLB_REFILL	-	-	-
0x00D4, IMP_L2D_S2_TLB	-	-	-
0x00D5, IMP_L2D_S2_TLB_REFILL	-	-	-
0x00D6, IMP_L2D_CACHE_STASH_DROPPED	-	-	-
0x00D7, IMP_L1D_TLB_REFILL_ETS	-	-	-

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x00E5, IMP_STALL_BACKEND_ILOCK_ADDR	-	-	-
0x00ED, IMP_STALL_BACKEND_BUSY_VPU_HAZARD	-	-	-
0x00EE, IMP_STALL_SLOT_BACKEND_ILOCK	-	-	• Stall
0x00F0, IMP_INST_SPEC_LDST_NUKE	-	-	-
0x00d9, IMP_L2D_CACHE_REFILL_HWPRF_CORRELATING	-	-	• L2_Cache
0x00da, IMP_L2D_CACHE_REFILL_HWPRF_SPATIAL	-	-	• L2_Cache
0x00db, IMP_L2D_CACHE_REFILL_HWPRF_OFFSET	-	-	• L2_Cache
0x00de, IMP_L3D_CACHE_HWPRF_STRIDE	-	-	• L3_Cache
0x00df, IMP_L3D_CACHE_HWPRF_OFFSET	-	-	• L3_Cache
0x00e6, IMP_STALL_BACKEND_ILOCK_VPU	• backend_stall_interlock_vpu_bound	• Topdown_Backend	• Stall
0x00ef, IMP_STALL_BACKEND_BUSY_VPU_ARB	• backend_busy_vpu_arb_bound	• Topdown_Backend	• Stall
0x00f1, IMP_STALL_BACKEND_BUSY_LS	• backend_busy_ls_bound	• Topdown_Backend	• Stall
0x00f2, IMP_STALL_BACKEND_ILOCK_LS	• backend_stall_interlock_ls_bound	• Topdown_Backend	• Stall
0x00f3, IMP_STALL_BACKEND_ILOCK_LS_INTO_AGU	• backend_stall_interlock_ptr_chase_bound	• Topdown_Backend	• Stall
0x00f4, IMP_STALL_SLOT_BACKEND_PORT_CONTENTION	-	-	• Stall
0x0100, IMP_L2D_CACHE_HIT_HWPRF	-	-	• L2_Cache
0x0101, IMP_L2D_CACHE_HIT_L1DHWPRF	-	-	• L2_Cache
0x0102, IMP_L2D_CACHE_HIT_L1DHWPRF_FHWPRF	-	-	• L2_Cache
0x0103, IMP_L2D_CACHE_HIT_PRF	-	-	• L2_Cache

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0104, IMP_L2D_CACHE_HIT_RD_FHWPRF_CORRELATING	-	-	• L2_Cache
0x0105, IMP_L2D_CACHE_HIT_RD_FHWPRF_OFFSET	-	-	• L2_Cache
0x0106, IMP_L2D_CACHE_HIT_RD_FHWPRF_SPATIAL	-	-	• L2_Cache
0x0107, IMP_L2D_CACHE_HIT_RD_FHWPRF_STRIDE	-	-	• L2_Cache
0x0108, IMP_L2D_CACHE_HIT_RD_FHWPRF_TLB	-	-	• L2_Cache
0x0109, IMP_L2D_CACHE_HWPRF_CORRELATING	-	-	• L2_Cache
0x010a, IMP_L2D_CACHE_HWPRF_OFFSET	-	-	• L2_Cache
0x010b, IMP_L2D_CACHE_HWPRF_SPATIAL	-	-	• L2_Cache
0x010c, IMP_L2D_CACHE_HWPRF_STRIDE	-	-	• L2_Cache
0x010d, IMP_L2D_CACHE_HWPRF _TLB	-	-	• L2_Cache
0x010e, IMP_L2D_CACHE_L1DHWPRF	-	-	• L2_Cache
0x010f, IMP_L2D_CACHE_REFILL_HWPRF_STRIDE	-	-	• L2_Cache
0x0110, IMP_L2D_CACHE_REFILL_HWPRF_TLB	-	-	• L2_Cache
0x0111, IMP_L2D_CACHE_REFILL_L1DHWPRF	-	-	• L2_Cache
0x0112, IMP_L2D_LFB_HIT_L1DHWPRF_FHWPRF	-	-	• L2_Cache
0x0120, IMP_L3D_CACHE_L1DHWPRF	-	-	• L3_Cache
0x0122, IMP_L3D_CACHE_REFILL_L1DHWPRF	-	-	• L3_Cache
0x0123, IMP_L3D_CACHE_REFILL_L2DHWPRF	-	-	• L3_Cache

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x0124, IMP_L3D_CACHE_REFILL_L2DHWPRF_CORRELATING	-	-	• L3_Cache
0x0125, IMP_L3D_CACHE_REFILL_L2DHWPRF_OFFSET	-	-	• L3_Cache
0x0126, IMP_L3D_CACHE_REFILL_L2DHWPRF_SPATIAL	-	-	• L3_Cache
0x0127, IMP_L3D_CACHE_REFILL_L2DHWPRF_STRIDE	-	-	• L3_Cache
0x0128, IMP_L3D_CACHE_REFILL_L2DHWPRF_TLB	-	-	• L3_Cache
0x3200, STALL_BACKEND_BUSY_CME	• backend_core_cme_bound • backend_cme_cpu_bound • backend_cme_backpressure_bound • backend_cme_busy_arb_bound	• Topdown_Backend • Topdown_CME	• Stall
0x3201, STALL_BACKEND_BUSY_CMEBOUND	• backend_cme_backpressure_bound	• Topdown_CME	• Stall
0x3202, STALL_BACKEND_BUSY_CME_ARB	• backend_cme_busy_arb_bound	• Topdown_CME	• Stall
0x3203, STALL_BACKEND_BUSY_CME_CPUBOUND	• backend_cme_cpu_bound	• Topdown_CME	• Stall
0x3204, STALL_BACKEND_ILOCK_FROM_CME	• backend_stall_interlock_from_cme_bound • backend_stall_interlock_from_cme_flags_bound • backend_stall_interlock_from_cme_gpr_bound • backend_stall_interlock_from_cme_predicate_bound	• CME_Ilock_From_CME • Topdown_Backend	• Stall
0x3205, STALL_BACKEND_ILOCK_FROM_CME_GPR	• backend_stall_interlock_from_cme_gpr_bound	• CME_Ilock_From_CME	• Stall
0x3206, STALL_BACKEND_ILOCK_FROM_CME_PRED	• backend_stall_interlock_from_cme_predicate_bound	• CME_Ilock_From_CME	• Stall
0x3207, STALL_BACKEND_ILOCK_FROM_CME_FLAGS	• backend_stall_interlock_from_cme_flags_bound	• CME_Ilock_From_CME	• Stall

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x3208, STALL_BACKEND_ILOCK_TO_CME	• backend_stall_interlock_to_cme_bound • backend_stall_interlock_to_cme_flags_bound • backend_stall_interlock_to_cme_gpr_bound • backend_stall_interlock_to_cme_predicate_bound	• CME_Ilock_To_CME • Topdown_Backend	• Stall
0x3209, STALL_BACKEND_ILOCK_TO_CME_GPR	• backend_stall_interlock_to_cme_gpr_bound	• CME_Ilock_To_CME	• Stall
0x320a, STALL_BACKEND_ILOCK_TO_CME_PRED	• backend_stall_interlock_to_cme_predicate_bound	• CME_Ilock_To_CME	• Stall
0x320b, STALL_BACKEND_ILOCK_TO_CME_FLAGS	• backend_stall_interlock_to_cme_flags_bound	• CME_Ilock_To_CME	• Stall
0x320c, STALL_BACKEND_MEM_CME_LSRT_FULL	• backend_mem_cme_lsrt_full_bound	• Topdown_Backend	• Stall
0x320d, STALL_BACKEND_MEM_CME_BARRIER	• backend_mem_cme_barrier_bound	• Topdown_Backend	• Stall
0x320e, STALL_BACKEND_MEM_CME_HZ_ON_CPU	• backend_mem_cme_hazard_cpu_bound	• Topdown_Backend	• Stall
0x320f, STALL_BACKEND_MEM_CPU_HZ_ON_CME	• backend_mem_cpu_hazard_cme_bound	• Topdown_Backend	• Stall
0x3210, STALL_BACKEND_MEM_CME	• backend_mem_cme_bound • backend_mem_cme_hazard_cpu_bound • backend_mem_cpu_hazard_cme_bound • backend_mem_cme_lsrt_full_bound • backend_mem_cme_barrier_bound	• Topdown_Backend	• Stall
0x3212, SM_ACTIVE_CYCLES	• sm_active_cycles_ratio	• Cycle_Accounting	• General
0x3213, CYCLES_CME_ALLOC	• cme_alloc_cycles_ratio	• Cycle_Accounting	• General
0x3214, CYCLES_ARB_PENDING_CME	• cme_arb_pending_ratio	• Cycle_Accounting	• General
0x3215, CYCLES_CME_RECONNECT_PENDING	-	-	• General
0x3216, ARB_CME_COUNT	-	-	• General
0x3217, RECONNECT_CME_COUNT	-	-	• General

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x3218, OP_CME_ISSUE	-	-	• Spec_Operation
0x3219, SSVE_INST_SPEC	-	-	• Spec_Operation
0x321a, SSVE_SPEC	-	-	• Spec_Operation
0x3234, SSVE_PRED_SPEC	-	-	• SVE
0x3235, SSVE_PRED_EMPTY_SPEC	-	-	• SVE
0x3236, SSVE_PRED_FULL_SPEC	-	-	• SVE
0x3237, SSVE_PRED_NOT_FULL _SPEC	-	-	• SVE
0x3238, SSVE_PRED_PARTIAL_SPEC	-	-	• SVE
0x4004, CNT_CYCLES	-	-	• General
0x4005, STALL_BACKEND_MEM	• backend_mem_cache_bound • backend_cache_l1d_bound • backend_cache_l2d_bound	• Topdown_Backend	• Stall
0x4006, L1I_CACHE_LMISS	-	-	• L1I_Cache
0x4009, L2D_CACHE_LMISS_RD	-	-	• L2_Cache
0x400B, L3D_CACHE_LMISS_RD	-	-	• L3_Cache
0x400C, TRB_WRAP	-	-	• Non_PMU
0x400D, PMU_OVFS	-	-	• Non_PMU
0x400E, TRB_TRIG	-	-	• Non_PMU
0x400F, PMU_HOVFS	-	-	• Non_PMU
0x4010, TRCEXTOUT0	-	-	• TRCEXT
0x4011, TRCEXTOUT1	-	-	• TRCEXT
0x4012, TRCEXTOUT2	-	-	• TRCEXT
0x4013, TRCEXTOUT3	-	-	• TRCEXT
0x4018, CTI_TRIGOUT4	-	-	• TRCEXT
0x4019, CTI_TRIGOUT5	-	-	• TRCEXT
0x401A, CTI_TRIGOUT6	-	-	• TRCEXT
0x401B, CTI_TRIGOUT7	-	-	• TRCEXT
0x4020, LDST_ALIGN_LAT	-	-	• Memory
0x4021, LD_ALIGN_LAT	-	-	• Memory
0x4022, ST_ALIGN_LAT	-	-	• Memory
0x4024, MEM_ACCESS_CHECKED	-	-	• Memory
0x4025, MEM_ACCESS_CHECKED_RD	-	-	• Memory
0x4026, MEM_ACCESS_CHECKED_WR	-	-	• Memory
0x8000, SIMD_INST_RETIRED	-	-	• Retired

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8004, SIMD_INST_SPEC	-	-	• Spec_Operation
0x8005, ASE_INST_SPEC	-	-	• Spec_Operation
0x8006, SVE_INST_SPEC	-	-	• Spec_Operation • SVE
0x8014, FP_HP_SPEC	• fp16_percentage	• FP_Precision_Mix	• SVE
0x8018, FP_SP_SPEC	• fp32_percentage	• FP_Precision_Mix	• SVE
0x801C, FP_DP_SPEC	• fp64_percentage	• FP_Precision_Mix	• SVE
0x8056, SVE_SPEC	• sve_percentage	• Operation_Mix	• Spec_Operation
0x8057, ASE_SVE_SPEC	-	-	• Spec_Operation
0x8074, SVE_PRED_SPEC	• sve_predicate_percentage • sve_predicate_full _percentage • sve_predicate_partial _percentage • sve_predicate_empty _percentage	• SVE_Effectiveness	• SVE
0x8075, SVE_PRED_EMPTY_SPEC	• sve_predicate_empty _percentage	• SVE_Effectiveness	• SVE
0x8076, SVE_PRED_FULL_SPEC	• sve_predicate_full _percentage	• SVE_Effectiveness	• SVE
0x8077, SVE_PRED_PARTIAL_SPEC	• sve_predicate_partial _percentage	• SVE_Effectiveness	• SVE
0x8078, SVE_UNPRED_SPEC	-	-	• SVE
0x8079, SVE_PRED_NOT_FULL_SPEC	-	-	• SVE
0x8080, SVE_LDST_SPEC	-	-	• Spec_Operation
0x8081, SVE_LD_SPEC	-	-	• Spec_Operation
0x8082, SVE_ST_SPEC	-	-	• Spec_Operation
0x8083, SVE_PRF_SPEC	-	-	• Spec_Operation
0x80BC, SVE_LDFF_SPEC	-	-	• SVE
0x80BD, SVE_LDFF_FAULT_SPEC	-	-	• SVE
0x80E3, ASE_SVE_INT8_SPEC	-	-	• SVE
0x80E7, ASE_SVE_INT16_SPEC	-	-	• SVE
0x80EB, ASE_SVE_INT32_SPEC	-	-	• SVE
0x80EF, ASE_SVE_INT64_SPEC	-	-	• SVE
0x8107, BR_SKIP_RETIRED	-	-	• Retired
0x8108, BR_IMMED_TAKEN_RETIRED	-	-	• Retired
0x810C, BR_INDNR_TAKEN_RETIRED	-	-	• Retired
0x8110, BR_IMMED_PRED_RETIRED	-	-	• Retired

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8111, BR_IMMED_MIS_PRED_RETIRED	-	-	• Retired
0x8112, BR_IND_PRED_RETIRED	-	-	• Retired
0x8113, BR_IND_MIS_PRED_RETIRED	-	-	• Retired
0x8114, BR_RETURN_PRED_RETIRED	-	-	• Retired
0x8115, BR_RETURN_MIS_PRED_RETIRED	-	-	• Retired
0x8116, BR_INDNR_PRED_RETIRED	-	-	• Retired
0x8117, BR_INDNR_MIS_PRED_RETIRED	-	-	• Retired
0x8118, BR_TAKEN_PRED_RETIRED	-	-	• Retired
0x8119, BR_TAKEN_MIS_PRED_RETIRED	-	-	• Retired
0x811A, BR_SKIP_PRED_RETIRED	-	-	• Retired
0x811B, BR_SKIP_MIS_PRED_RETIRED	-	-	• Retired
0x811C, BR_PRED_RETIRED	-	-	• Retired
0x811D, BR_IND_RETIRED	• branch_indirect_ratio	• Branch_Effectiveness	• Retired
0x8120, INST_FETCH_PERCYC	• instruction_fetch_average_latency	• Average_Latency	• Memory
0x8121, MEM_ACCESS_RD_PERCYC	• load_average_latency	• Average_Latency	• Memory
0x8124, INST_FETCH	• instruction_fetch_average_latency	• Average_Latency	• Memory
0x8125, BUS_REQ_RD_PERCYC	• bus_read_requests_average_latency in Average_Latency • bus_read_requests_average_latency in Bus_Effectiveness	• Average_Latency • Bus_Effectiveness	• Bus
0x8128, DTLB_WALK_PERCYC	• dtlb_walk_average_latency in Average_Latency • dtlb_walk_average_latency in DTLB_Effectiveness	• Average_Latency • DTLB_Effectiveness	• TLB
0x8129, ITLB_WALK_PERCYC	• itlb_walk_average_latency in Average_Latency • itlb_walk_average_latency in ITLB_Effectiveness	• Average_Latency • ITLB_Effectiveness	• TLB
0x8131, L1I_TLB_RD	-	-	• TLB
0x8134, DTLB_HWUPD	-	-	• TLB
0x8135, ITLB_HWUPD	-	-	• TLB

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8136, DTLB_STEP	• dtlb_walk_average_depth	• DTLB_Effectiveness	• TLB
0x8137, ITLB_STEP	• itlb_walk_average_depth	• ITLB_Effectiveness	• TLB
0x8138, DTLB_WALK_LARGE	-	-	• TLB
0x8139, ITLB_WALK_LARGE	-	-	• TLB
0x813A, DTLB_WALK_SMALL	-	-	• TLB
0x813B, ITLB_WALK_SMALL	-	-	• TLB
0x813C, DTLB_WALK_RW	-	-	• TLB
0x813D, ITLB_WALK_RD	-	-	• TLB
0x8140, L1D_CACHE_RW	• l1d_cache_miss_ratio in L1D_Cache _Effectiveness • l1d_cache_miss_ratio in Miss_Ratio	• L1D_Cache _Effectiveness • Miss_Ratio	• L1D_Cache
0x8141, L1I_CACHE_RD	-	-	• L1I_Cache
0x8142, L1D_CACHE_PRFM	-	-	• L1D_Cache
0x8145, L1I_CACHE_HWPRF	-	-	• L1I_Cache
0x8146, L1D_CACHE_REFILL_PRFM	-	-	• L1D_Cache
0x8148, L2D_CACHE_RW	• l2d_cache_miss_ratio in L2D_Cache _Effectiveness • l2d_cache_miss_ratio in Miss_Ratio	• L2D_Cache _Effectiveness • Miss_Ratio	• L2_Cache
0x8149, L2I_CACHE_RD	-	-	• L2_Cache
0x814A, L2D_CACHE_PRFM	-	-	• L2_Cache
0x814E, L2D_CACHE_REFILL_PRFM	-	-	• L2_Cache
0x8151, L3D_CACHE_PRFM	-	-	• L3_Cache
0x8154, L1D_CACHE_HWPRF	-	-	• L1D_Cache
0x8155, L2D_CACHE_HWPRF	-	-	• L2_Cache
0x8156, L3D_CACHE_HWPRF	-	-	• L3_Cache
0x8157, LL_CACHE_HWPRF	-	-	• LL_Cache
0x8158, STALL_FRONTEND _MEMBOUND	• frontend_mem_bound • frontend_mem_cache_bound • frontend_mem_tlb_bound	• Topdown_Frontend	• Stall
0x8159, STALL_FRONTEND_L1I	• frontend_mem_cache_bound • frontend_cache_l1i_bound • frontend_cache_l2i_bound	• Topdown_Frontend	• Stall
0x815B, STALL_FRONTEND_MEM	• frontend_mem_cache_bound • frontend_cache_l1i_bound • frontend_cache_l2i_bound	• Topdown_Frontend	• Stall
0x815C, STALL_FRONTEND_TLB	• frontend_mem_tlb_bound	• Topdown_Frontend	• Stall

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8160, STALL_FRONTEND _CPUBOUND	• frontend_core_bound • frontend_core_flush_bound • frontend_core_flow_bound	• Topdown_Frontend	• Stall
0x8161, STALL_FRONTEND_FLOW	• frontend_core_flow_bound	• Topdown_Frontend	• Stall
0x8162, STALL_FRONTEND_FLUSH	• frontend_bound • bad_speculation • frontend_core_flush_bound	• Topdown_Frontend • Topdown_L1	• Stall
0x8164, STALL_BACKEND_MEMBOUND	• backend_mem_bound • backend_mem_cache_bound • backend_mem_tlb_bound • backend_mem_store_bound • backend_mem_cme_bound	• Topdown_Backend	• Stall
0x8165, STALL_BACKEND_L1D	• backend_mem_cache_bound • backend_cache_l1d_bound • backend_cache_l2d_bound	• Topdown_Backend	• Stall
0x8167, STALL_BACKEND_TLB	• backend_mem_tlb_bound	• Topdown_Backend	• Stall
0x8168, STALL_BACKEND_ST	• backend_mem_store_bound	• Topdown_Backend	• Stall
0x816A, STALL_BACKEND_CPUBOUND	• backend_core_bound • backend_core_cme_bound	• Topdown_Backend	• Stall
0x816B, STALL_BACKEND_BUSY	• backend_busy_bound	• Topdown_Backend	• Stall
0x816C, STALL_BACKEND_ILOCK	• backend_stall_interlock _bound • backend_stall_interlock _from_cme_bound • backend_stall_interlock _to_cme_bound	• Topdown_Backend	• Stall
0x816E, STALL_BACKEND_ATOMIC	-	-	• Stall
0x816F, STALL_BACKEND _MEMCPYSET	-	-	• Stall
0x8170, CAS_NEAR_FAIL	-	-	• Spec_Operation
0x8171, CAS_NEAR_PASS	• cas_near_pass_ratio	• Atomics _Effectiveness	• Spec_Operation
0x8172, CAS_NEAR_SPEC	• cas_near_ratio • cas_far_ratio • cas_near_pass_ratio	• Atomics _Effectiveness	• Spec_Operation
0x8173, CAS_FAR_SPEC	-	-	• Spec_Operation
0x8174, CAS_SPEC	• cas_near_ratio • cas_far_ratio	• Atomics _Effectiveness	• Spec_Operation
0x8175, LSE_LD_SPEC	• lse_load_ratio	• Atomics _Effectiveness	• Spec_Operation
0x8176, LSE_ST_SPEC	• lse_store_ratio	• Atomics _Effectiveness	• Spec_Operation

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8177, LSE_LDST_SPEC	• lse_atomics_ratio • lse_load_ratio • lse_store_ratio	• Atomics _Effectiveness	• Spec_Operation
0x8179, BRNL_INDNR_TAKEN _RETIRED	-	-	• Retired
0x817A, BL_TAKEN_RETIRED	-	-	• Retired
0x817B, BRNL_TAKEN_RETIRED	-	-	• Retired
0x817C, BL_IND_TAKEN_RETIRED	-	-	• Retired
0x817D, BRNL_IND_TAKEN_RETIRED	-	-	• Retired
0x817E, BL_IMMED_TAKEN_RETIRED	-	-	• Retired
0x817F, BRNL_IMMED_TAKEN _RETIRED	-	-	• Retired
0x8180, BR_UNCOND_RETIRED	-	-	• Retired
0x8181, BR_COND_RETIRED	-	-	• Retired
0x8182, BR_COND_TAKEN_RETIRED	-	-	• Retired
0x8183, BR_HINT_COND_RETIRED	-	-	• Retired
0x8184, BR_HINT_COND_PRED _RETIRED	-	-	• Retired
0x8185, BR_HINT_COND_MIS_PRED _RETIRED	-	-	• Retired
0x8186, UOP_RETIRED	-	-	• Retired
0x8188, DTLB_WALK_BLOCK	• dtlb_walk_block_ratio	• DTLB_Effectiveness	• TLB
0x8189, ITLB_WALK_BLOCK	• itlb_walk_block_ratio	• ITLB_Effectiveness	• TLB
0x818A, DTLB_WALK_PAGE	• dtlb_walk_page_ratio	• DTLB_Effectiveness	• TLB
0x818B, ITLB_WALK_PAGE	• itlb_walk_page_ratio	• ITLB_Effectiveness	• TLB
0x818D, BUS_REQ_RD	• bus_read_requests_average _latency in Average_Latency • bus_read_requests_average _latency in Bus_Effectiveness	• Average_Latency • Bus_Effectiveness	• Bus
0x818E, BUS_REQ_WR	-	-	• Bus
0x818F, BUS_REQ	-	-	• Bus
0x8190, ISNP_HIT_RD	• system_peer_cluster_cache _hit_ratio	• System_Memory _Effectiveness	• Coherency
0x8191, ISNP_HIT_NEAR_RD	-	-	• Coherency
0x8192, ISNP_HIT_FAR_RD	-	-	• Coherency
0x8194, DSNP_HIT_RD	-	-	• Coherency

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8195, DSNP_HIT_NEAR_RD	-	-	• Coherency
0x8196, DSNP_HIT_FAR_RD	-	-	• Coherency
0x8198, DSNP_HIT_WR	-	-	• Coherency
0x8199, DSNP_HIT_NEAR_WR	-	-	• Coherency
0x819A, DSNP_HIT_FAR_WR	-	-	• Coherency
0x819C, DSNP_HIT_RW	-	-	• Coherency
0x819D, DSNP_HIT_NEAR_RW	-	-	• Coherency
0x819E, DSNP_HIT_FAR_RW	-	-	• Coherency
0x81B4, DSNP_HIT	• system_peer_cluster_cache _hit_ratio	• System_Memory _Effectiveness	• Coherency
0x81B5, DSNP_HIT_NEAR	-	-	• Coherency
0x81B6, DSNP_HIT_FAR	-	-	• Coherency
0x81BC, L1D_CACHE_REFILL_HWPRF	-	-	• L1D_Cache
0x81BD, L2D_CACHE_REFILL_HWPRF	• l2_prefetcher_accuracy _l1hwprf_exclusive	• Prefetcher _Effectiveness	• L2_Cache
0x81C0, L1I_CACHE_HIT_RD	-	-	• L1I_Cache
0x81C1, L2I_CACHE_HIT_RD	-	-	• L2_Cache
0x81C4, L1D_CACHE_HIT_RD	-	-	• L1D_Cache
0x81C5, L2D_CACHE_HIT_RD	-	-	• L2_Cache
0x81C6, L3D_CACHE_HIT_RD	• system_l3_cache_hit_ratio	• System_Memory _Effectiveness	• L3_Cache
0x81C8, L1D_CACHE_HIT_WR	-	-	• L1D_Cache
0x81C9, L2D_CACHE_HIT_WR	-	-	• L2_Cache
0x81CC, L1D_CACHE_HIT_RW	-	-	• L1D_Cache
0x81CD, L2D_CACHE_HIT_RW	-	-	• L2_Cache
0x81E5, L2D_CACHE_HIT_RD _FHWPRF	-	-	• L2_Cache
0x81E6, L3D_CACHE_HIT_RD _FHWPRF	-	-	• L3_Cache
0x81E9, L2D_CACHE_HIT_WR _FHWPRF	-	-	• L2_Cache
0x81ED, L2D_CACHE_HIT_RW _FHWPRF	• l2_prefetcher_coverage _l1hwprf_exclusive • l2_prefetcher_accuracy _l1hwprf_exclusive • l2_prefetcher_timeliness _l1hwprf_exclusive	• Prefetcher _Effectiveness	• L2_Cache
0x8200, L1I_CACHE_HIT	-	-	• L1I_Cache
0x8201, L2I_CACHE_HIT	-	-	• L2_Cache
0x8204, L1D_CACHE_HIT	-	-	• L1D_Cache
0x8205, L2D_CACHE_HIT	-	-	• L2_Cache
0x820D, L2D_CACHE_HIT_PRFM	-	-	• L2_Cache

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8240, L1I_LFB_HIT_RD	-	-	• L1I_Cache
0x8244, L1D_LFB_HIT_RD	-	-	• L1D_Cache
0x8245, L2D_LFB_HIT_RD	-	-	• L2_Cache
0x8248, L1D_LFB_HIT_WR	-	-	• L1D_Cache
0x8249, L2D_LFB_HIT_WR	-	-	• L2_Cache
0x824C, L1D_LFB_HIT_RW	-	-	• L1D_Cache
0x824D, L2D_LFB_HIT_RW	-	-	• L2_Cache
0x8264, L1D_LFB_HIT_RD_FHWPRF	-	-	• L1D_Cache
0x8265, L2D_LFB_HIT_RD_FHWPRF	-	-	• L2_Cache
0x8268, L1D_LFB_HIT_WR_FHWPRF	-	-	• L1D_Cache
0x8269, L2D_LFB_HIT_WR_FHWPRF	-	-	• L2_Cache
0x826C, L1D_LFB_HIT_RW_FHWPRF	-	-	• L1D_Cache
0x826D, L2D_LFB_HIT_RW_FHWPRF	• l2_prefetcher_coverage _l1hwprf_exclusive • l2_prefetcher_accuracy _l1hwprf_exclusive • l2_prefetcher_timeliness _l1hwprf_exclusive	• Prefetcher _Effectiveness	• L2_Cache
0x8284, L1D_CACHE_PRF	-	-	• L1D_Cache
0x8285, L2D_CACHE_PRF	-	-	• L2_Cache
0x8286, L3D_CACHE_PRF	-	-	• L3_Cache
0x8287, LL_CACHE_PRF	-	-	• LL_Cache
0x828C, L1D_CACHE_REFILL_PRF	-	-	• L1D_Cache
0x828D, L2D_CACHE_REFILL_PRF	-	-	• L2_Cache
0x8298, LL_CACHE_RW	-	-	• LL_Cache
0x8299, LL_CACHE_PRFM	-	-	• LL_Cache
0x82A0, MEM_ACCESS_RW	-	-	• Memory
0x82A1, INST_FETCH_RD	-	-	• Memory
0x82FA, DTLB_WALK_HWPRF	-	-	• TLB
0x82FE, L1D_TLB_HWPRF	-	-	• TLB
0x835D, SE_SPEC	-	-	• Spec_Operation
0x835E, SME_INST_SPEC	• sme_percentage	• Operation_Mix	• Spec_Operation
0x835F, SE_INST_SPEC	-	-	• Spec_Operation
0x8380, ZA_ACTIVE	• za_active_cycles_ratio	• Cycle_Accounting	• General

5. Metrics by metric group in C1-Nano

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: 119

Metrics for C1-Nano are grouped into the following metric groups:

- Topdown_Backend, Topdown Backend (22 metrics)
- CME_Ilock_To_CME, Interlock to SME2 (3 metrics)
- CME_Ilock_From_CME, Interlock from SME2 (3 metrics)
- Cycle_Accounting, Cycle Accounting (6 metrics)
- Topdown_CME, Topdown SME2 (3 metrics)
- Topdown_L1, Topdown Level 1 (4 metrics)
- Topdown_Frontend, Topdown Frontend (8 metrics)
- General, General (1 metrics)
- MPKI, Misses Per Kilo Instructions (12 metrics)
- Miss_Ratio, Miss Ratio (12 metrics)
- SVE_Effectiveness, SVE Effectiveness (4 metrics)
- FP_Precision_Mix, Floating Point Precision (3 metrics)
- Branch_Effectiveness, Branch Effectiveness (5 metrics)
- ITLB_Effectiveness, Instruction TLB Effectiveness (10 metrics)
- DTLB_Effectiveness, Data TLB Effectiveness (10 metrics)
- L1I_Cache_Effectiveness, L1 Instruction Cache Effectiveness (2 metrics)
- L1D_Cache_Effectiveness, L1 Data Cache Effectiveness (2 metrics)
- L2I_Cache_Effectiveness, L2I Unified Cache Effectiveness (2 metrics)
- L2D_Cache_Effectiveness, L2D Unified Cache Effectiveness (2 metrics)
- L3_Cache_Effectiveness, L3 Unified Cache Effectiveness (2 metrics)
- LL_Cache_Effectiveness, Last Level Cache Effectiveness (3 metrics)
- Operation_Mix, Speculative Operation Mix (12 metrics)
- Prefetcher_Effectiveness, Prefetcher Effectiveness (3 metrics)
- Average_Latency, Average Latency (5 metrics)
- Bus_Effectiveness, Bus Effectiveness (1 metrics)

- System_Memory_Effectiveness, System Memory Effectiveness (2 metrics)
- Atomics_Effectiveness, Atomics Effectiveness (6 metrics)

5.1 Topdown_Backend metrics for C1-Nano

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

Summary of metrics in Topdown_Backend:

- Total metrics: 22

Table 5-1: Topdown_Backend metrics summary

Metric	Name	Description
backend_busy_bound	Backend Busy Bound	This metric is the percentage of total cycles stalled in the backend due to issue queues being...
backend_busy_ls_bound	Backend Busy LS Bound	This metric is the percentage of total cycles stalled in the backend due to memory system issue...
backend_busy_vpu_arb_bound	Backend Busy VPU Arbitration Bound	This metric is the percentage of total cycles stalled in the backend due to VPU arbitration...
backend_cache_l1d_bound	Backend Cache L1D Bound	This metric is the percentage of total cycles stalled in the backend due to memory access latency...
backend_cache_l2d_bound	Backend Cache L2D Bound	This metric is the percentage of total cycles stalled in the backend due to memory access latency...
backend_core_bound	Backend Core Bound	This metric is the percentage of total cycles stalled in the backend due to backend core resource...
backend_core_cme_bound	Backend Core SME2 Bound	This metric is the percentage of total cycles stalled in the backend due to the resource...
backend_mem_bound	Backend Memory Bound	This metric is the percentage of total cycles stalled in the backend due to backend core resource...
backend_mem_cache_bound	Backend Mem Cache Bound	This metric is the percentage of total cycles stalled in the backend due to memory latency issues...
backend_mem_cme_barrier_bound	Backend SME2 LSRT Barrier Bound	This metric is the percentage of total cycles stalled in the backend as the SME2 unit is...
backend_mem_cme_bound	Backend Memory SME2 Bound	This metric is the percentage of total cycles stalled in the backend caused by load or store...
backend_mem_cme_hazard_cpu_bound	Backend SME2 Memory Hazard CPU Bound	This metric is the percentage of total cycles stalled in the backend due to a SME2 LSRT hazard...
backend_mem_cme_lsrt_full_bound	Backend SME2 LSRT Full Bound	This metric is the percentage of total cycles stalled in the backend as the SME2 unit is busy due...
backend_mem_cpu_hazard_cme_bound	Backend CPU Memory Hazard SME2 Bound	This metric is the percentage of total cycles stalled in the backend as the SME2 unit is busy due...
backend_mem_store_bound	Backend Mem Store Bound	This metric is the percentage of total cycles stalled in the backend due to memory write pending...
backend_mem_tlb_bound	Backend Mem Tlb Bound	This metric is the percentage of total cycles stalled in the backend due to memory access latency...

Metric	Name	Description
backend_stall_interlock_bound	Backend Stall Interlock Rate	This metric is the percentage of total cycles stalled in the backend due to instruction...
backend_stall_interlock_from_cme_bound	Backend ILOCK From SME2 Bound	This metric is the percentage of total cycles stalled in the backend when a CPU instruction is...
backend_stall_interlock_ls_bound	Backend Stall memory source interlock	This metric is the percentage of backend stalls due to an interlock, where the source of at least...
backend_stall_interlock_ptr_chase_bound	Backend Stall pointer chase interlock Rate	This metric is the percentage of backend stalls due to a pointer chase interlock - that is, the...
backend_stall_interlock_to_cme_bound	Backend ILOCK To SME2 Bound	This metric is the percentage of total cycles stalled in the backend when the oldest...
backend_stall_interlock_vpu_bound	Backend Stall VPU source interlock	This metric is the percentage of backend stalls due to an interlock, where the source of at least...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

backend_busy_bound, Backend Busy Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_BUSY / STALL_BACKEND * 100

Related telemetry artifacts

Events

STALL_BACKEND
STALL_BACKEND_BUSY

Metric group

Topdown_Backend

Methodology

Stage 1

backend_busy_ls_bound, Backend Busy LS Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

IMP_STALL_BACKEND_BUSY_LS / STALL_BACKEND * 100

Related telemetry artifacts

Events

IMP_STALL_BACKEND_BUSY_LS
STALL_BACKEND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_busy_vpu_arb_bound, Backend Busy VPU Arbitration Bound, metric

This metric is the percentage of total cycles stalled in the backend due to VPU arbitration preventing the VPU from accepting operations for execution.

Units

This unit is expressed as percent of cycles.

Formula

IMP_STALL_BACKEND_BUSY_VPU_ARB / STALL_BACKEND * 100

Related telemetry artifacts

Events

IMP_STALL_BACKEND_BUSY_VPU_ARB
STALL_BACKEND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_cache_l1d_bound, Backend Cache L1D Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_L1D / (STALL_BACKEND_L1D + STALL_BACKEND_MEM) * 100

Related telemetry artifacts

Events

STALL_BACKEND_L1D
STALL_BACKEND_MEM

Metric group

Topdown_Backend

Methodology

Stage 1

backend_cache_l2d_bound, Backend Cache L2D Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_MEM / (STALL_BACKEND_L1D + STALL_BACKEND_MEM) * 100

Related telemetry artifacts

Events

STALL_BACKEND_L1D
STALL_BACKEND_MEM

Metric group

Topdown_Backend

Methodology

Stage 1

backend_core_bound, Backend Core Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_CPUBOUND / STALL_BACKEND * 100

Related telemetry artifacts

Events

STALL_BACKEND
STALL_BACKEND_CPUBOUND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_core_cme_bound, Backend Core SME2 Bound, metric

This metric is the percentage of total cycles stalled in the backend due to the resource constraints to dispatch to the SME2 unit.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_BUSY_CME / STALL_BACKEND_CPUBOUND * 100

Related telemetry artifacts

Events

STALL_BACKEND_BUSY_CME
STALL_BACKEND_CPUBOUND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_bound, Backend Memory Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_MEMBOUND / STALL_BACKEND * 100

Related telemetry artifacts

Events

STALL_BACKEND
STALL_BACKEND_MEMBOUND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_cache_bound, Backend Mem Cache Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

(STALL_BACKEND_L1D + STALL_BACKEND_MEM) / STALL_BACKEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_BACKEND_L1D
STALL_BACKEND_MEM

STALL_BACKEND_MEMBOUND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_cme_barrier_bound, Backend SME2 LSRT Barrier Bound, metric

This metric is the percentage of total cycles stalled in the backend as the SME2 unit is busy because a CPU barrier waits for SME2 load/store transaction completion.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_MEM_CME_BARRIER / STALL_BACKEND_MEM_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_MEM_CME
STALL_BACKEND_MEM_CME_BARRIER

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_cme_bound, Backend Memory SME2 Bound, metric

This metric is the percentage of total cycles stalled in the backend caused by load or store address hazards caused by SME2 unit memory execution dependency. SME2 unit and PE share the CPU memory subsystem and are synchronized via the LSRT block, causing memory execution units in them to have dependencies on data accesses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_MEM_CME / STALL_BACKEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_BACKEND_MEMBOUND
STALL_BACKEND_MEM_CME

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_cme_hazard_cpu_bound, Backend SME2 Memory Hazard CPU Bound, metric

This metric is the percentage of total cycles stalled in the backend due to a SME2 LSRT hazard causing SME2 instructions to stall. SME2 unit and PE share the CPU memory subsystem and are synchronized via the LSRT block, causing both the execution units to have dependencies on data accesses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_MEM_CME_HZ_ON_CPU / STALL_BACKEND_MEM_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_MEM_CME
STALL_BACKEND_MEM_CME_HZ_ON_CPU

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_cme_lsrt_full_bound, Backend SME2 LSRT Full Bound, metric

This metric is the percentage of total cycles stalled in the backend as the SME2 unit is busy due to the LSRT being full. SME2 unit and PE share the CPU memory subsystem and are synchronized via the LSRT block, causing both execution units to have dependencies on data accesses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_MEM_CME_LSRT_FULL / STALL_BACKEND_MEM_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_MEM_CME
STALL_BACKEND_MEM_CME_LSRT_FULL

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_cpu_hazard_cme_bound, Backend CPU Memory Hazard SME2 Bound, metric

This metric is the percentage of total cycles stalled in the backend as the SME2 unit is busy due to a SME2 LSRT hazard causing CPU instructions to stall. SME2 unit and PE share the CPU

memory subsystem and are synchronized via the LSRT block, causing both execution units to have dependencies on data accesses.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_MEM_CPU_HZ_ON_CME / STALL_BACKEND_MEM_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_MEM_CME
STALL_BACKEND_MEM_CPU_HZ_ON_CME

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_store_bound, Backend Mem Store Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_ST / STALL_BACKEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_BACKEND_MEMBOUND
STALL_BACKEND_ST

Metric group

Topdown_Backend

Methodology

Stage 1

backend_mem_tlb_bound, Backend Mem Tlb Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_TLB / STALL_BACKEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_BACKEND_MEMBOUND
STALL_BACKEND_TLB

Metric group

Topdown_Backend

Methodology

Stage 1

backend_stall_interlock_bound, Backend Stall Interlock Rate, metric

This metric is the percentage of total cycles stalled in the backend due to instruction interlocks. A high rate of interlock stalls is likely an indication of sub-optimal instruction scheduling for an in-order core.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_ILOCK / STALL_BACKEND * 100

Related telemetry artifacts

Events

STALL_BACKEND
STALL_BACKEND_ILOCK

Metric group

Topdown_Backend

Methodology

Stage 1

backend_stall_interlock_from_cme_bound, Backend ILOCK From SME2 Bound, metric

This metric is the percentage of total cycles stalled in the backend when a CPU instruction is stalled on the SME2 unit due to a data hazard between the SME2 unit and the CPU, including predicate, flag and GPR updates.

Units

This unit is expressed as percent of ilock cycles.

Formula

STALL_BACKEND_ILOCK_FROM_CME / STALL_BACKEND_ILOCK * 100

Related telemetry artifacts

Events

STALL_BACKEND_ILOCK
STALL_BACKEND_ILOCK_FROM_CME

Metric group

Topdown_Backend

Methodology

Stage 1

backend_stall_interlock_ls_bound, Backend Stall memory source interlock, metric

This metric is the percentage of backend stalls due to an interlock, where the source of at least one interlock is a memory operation. Refer to `backend_mem_bound` and subsequent breakdown to check whether the interlock is likely affected by memory resources such as caches or TLBs, or is instead primarily affected by load-to-use latency.

Units

This unit is expressed as percent of cycles.

Formula

IMP_STALL_BACKEND_ILOCK_LS / STALL_BACKEND * 100

Related telemetry artifacts

Events

IMP_STALL_BACKEND_ILOCK_LS
STALL_BACKEND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_stall_interlock_ptr_chase_bound, Backend Stall pointer chase interlock Rate, metric

This metric is the percentage of backend stalls due to a pointer chase interlock - that is, the source of at least one interlock is a memory instruction, and the destination is the address of another memory instruction. Refer to `backend_mem_bound` and subsequent breakdown to check whether the interlock is likely affected by memory resources such as caches or TLBs, or is instead primarily affected by load-to-use latency.

Units

This unit is expressed as percent of cycles.

Formula

IMP_STALL_BACKEND_ILOCK_LS_INTO_AGU / STALL_BACKEND * 100

Related telemetry artifacts

Events

IMP_STALL_BACKEND_ILOCK_LS_INTO_AGU
STALL_BACKEND

Metric group

Topdown_Backend

Methodology

Stage 1

backend_stall_interlock_to_cme_bound, Backend ILOCK To SME2 Bound, metric

This metric is the percentage of total cycles stalled in the backend when the oldest SME2 instruction is stalled due to a data hazard between the SME2 unit and the CPU, including predicate, flag and GPR updates.

Units

This unit is expressed as percent of ilock cycles.

Formula

STALL_BACKEND_ILOCK_TO_CME / STALL_BACKEND_ILOCK * 100

Related telemetry artifacts

Events

STALL_BACKEND_ILOCK
STALL_BACKEND_ILOCK_TO_CME

Metric group

Topdown_Backend

Methodology

Stage 1

backend_stall_interlock_vpu_bound, Backend Stall VPU source interlock, metric

This metric is the percentage of backend stalls due to an interlock, where the source of at least one interlock is a VPU operation.

Units

This unit is expressed as percent of cycles.

Formula

IMP_STALL_BACKEND_ILOCK_VPU / STALL_BACKEND * 100

Related telemetry artifacts

Events

IMP_STALL_BACKEND_ILOCK_VPU
STALL_BACKEND

Metric group

Topdown_Backend

Methodology

Stage 1

5.2 CME_Ilock_To_CME metrics for C1-Nano

Interlock to SME2. This metric group contains a set of metrics that measure the percentage of processor cycles stalled in backend of the processor due to the oldest SME2 instruction being stalled due to a data hazard.

Summary of metrics in CME_Ilock_To_CME:

- Total metrics: 3

Table 5-2: CME_Ilock_To_CME metrics summary

Metric	Name	Description
backend_stall_interlock_to_cme_flags_bound	Backend ILOCK To SME2 Flags Bound	This metric is the percentage of total cycles stalled in the backend when the oldest...
backend_stall_interlock_to_cme_gpr_bound	Backend ILOCK To SME2 GPR Bound	This metric is the percentage of total cycles stalled in the backend when the oldest...
backend_stall_interlock_to_cme_predicate_bound	Backend ILOCK To SME2 Predicate Bound	This metric is the percentage of total cycles stalled in the backend when the oldest...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

backend_stall_interlock_to_cme_flags_bound, Backend ILOCK To SME2 Flags Bound, metric

This metric is the percentage of total cycles stalled in the backend when the oldest SME2 instruction is stalled due to a data hazard between the SME2 unit and the CPU, where at least one instruction is waiting for flags produced by the CPU.

Units

This unit is expressed as percent of ilock cycles to cme.

Formula

STALL_BACKEND_ILOCK_TO_CME_FLAGS / STALL_BACKEND_ILOCK_TO_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_ILOCK_TO_CME
STALL_BACKEND_ILOCK_TO_CME_FLAGS

Metric group

CME_Ilock_To_CME

Methodology

Stage 2

backend_stall_interlock_to_cme_gpr_bound, Backend ILOCK To SME2 GPR Bound, metric

This metric is the percentage of total cycles stalled in the backend when the oldest SME2 instruction is stalled due to a data hazard between the SME2 unit and the CPU, where at least one instruction is waiting for GPR produced by the CPU.

Units

This unit is expressed as percent of ilock cycles to cme.

Formula

STALL_BACKEND_ILOCK_TO_CME_GPR / STALL_BACKEND_ILOCK_TO_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_ILOCK_TO_CME
STALL_BACKEND_ILOCK_TO_CME_GPR

Metric group

CME_Ilock_To_CME

Methodology

Stage 2

backend_stall_interlock_to_cme_predicate_bound, Backend ILOCK To SME2 Predicate Bound, metric

This metric is the percentage of total cycles stalled in the backend when the oldest SME2 instruction is stalled due to a data hazard between the SME2 unit and the CPU, where at least one instruction is waiting for predicate bound produced by the CPU.

Units

This unit is expressed as percent of ilock cycles to cme.

Formula

STALL_BACKEND_ILOCK_TO_CME_PRED / STALL_BACKEND_ILOCK_TO_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_ILOCK_TO_CME
STALL_BACKEND_ILOCK_TO_CME_PRED

Metric group

CME_Ilock_To_CME

Methodology

Stage 2

5.3 CME_Ilock_From_CME metrics for C1-Nano

Interlock from SME2. This metric group contains a set of metrics that measure the percentage of processor cycles stalled in backend of the processor due to CPU instruction being stalled due to a data hazard from SME2.

Summary of metrics in CME_Ilock_From_CME:

- Total metrics: 3

Table 5-3: CME_Ilock_From_CME metrics summary

Metric	Name	Description
backend_stall_interlock_from_cme_flags_bound	Backend ILOCK From SME2 Flags Bound	This metric is the percentage of total cycles stalled in the backend when a CPU instruction is...
backend_stall_interlock_from_cme_gpr_bound	Backend ILOCK From SME2 GPR Bound	This metric is the percentage of total cycles stalled in the backend when a CPU instruction is...
backend_stall_interlock_from_cme_predicate_bound	Backend ILOCK From SME2 Predicate Bound	This metric is the percentage of total cycles stalled in the backend when a CPU instruction is...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

backend_stall_interlock_from_cme_flags_bound, Backend ILOCK From SME2 Flags Bound, metric

This metric is the percentage of total cycles stalled in the backend when a CPU instruction is stalled on the SME2 unit due to a data hazard between the SME2 unit and the CPU, where at least one instruction is waiting for flags produced by the CPU.

Units

This unit is expressed as percent of ilock cycles from cme.

Formula

STALL_BACKEND_ILOCK_FROM_CME_FLAGS / STALL_BACKEND_ILOCK_FROM_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_ILOCK_FROM_CME
STALL_BACKEND_ILOCK_FROM_CME_FLAGS

Metric group

CME_Ilock_From_CME

Methodology

Stage 2

backend_stall_interlock_from_cme_gpr_bound, Backend ILOCK From SME2 GPR Bound, metric

This metric is the percentage of total cycles stalled in the backend when a CPU instruction is stalled on the SME2 unit due to a data hazard between the SME2 unit and the CPU, where at least one instruction is waiting for GPR produced by the CPU.

Units

This unit is expressed as percent of ilock cycles from cme.

Formula

STALL_BACKEND_ILOCK_FROM_CME_GPR / STALL_BACKEND_ILOCK_FROM_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_ILOCK_FROM_CME
STALL_BACKEND_ILOCK_FROM_CME_GPR

Metric group

CME_Ilock_From_CME

Methodology

Stage 2

backend_stall_interlock_from_cme_predicate_bound, Backend ILOCK From SME2 Predicate Bound, metric

This metric is the percentage of total cycles stalled in the backend when a CPU instruction is stalled on the SME2 unit due to a data hazard between the SME2 unit and the CPU, where at least one instruction is waiting for predicate produced by the CPU.

Units

This unit is expressed as percent of ilock cycles from cme.

Formula

STALL_BACKEND_ILOCK_FROM_CME_PRED / STALL_BACKEND_ILOCK_FROM_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_ILOCK_FROM_CME
STALL_BACKEND_ILOCK_FROM_CME_PRED

Metric group

CME_Ilock_From_CME

Methodology

Stage 2

5.4 Cycle_Accounting metrics for C1-Nano

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: 6

Table 5-4: Cycle_Accounting metrics summary

Metric	Name	Description
backend_stalled_cycles	Backend Stalled Cycles	This metric is the percentage of cycles that were stalled due to resource constraints in the...
cme_alloc_cycles_ratio	SME2 Allocation Cycles Ratio	This metric is measures the ratio of cycles where the CPU had an SME2 unit allocated to it, such...
cme_arb_pending_ratio	SME2 Arbitration Pending Cycles Ratio	This metric is measures the ratio of cycles where the CPU is in arbitration while attempting to...
frontend_stalled_cycles	Frontend Stalled Cycles	This metric is the percentage of cycles that were stalled due to resource constraints in the...
sm_active_cycles_ratio	SM Active Cycles Ratio	This metric is measures the ratio of cycles when PSTATE.SM was enabled to the total number of CPU...
za_active_cycles_ratio	ZA Active Cycles Ratio	This metric is measures the ratio of cycles when PSTATE.ZA was enabled to the total number of CPU...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

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

STALL_BACKEND / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
STALL_BACKEND

Metric group

Cycle_Accounting

Methodology

Stage 2

cme_alloc_cycles_ratio, SME2 Allocation Cycles Ratio, metric

This metric is measures the ratio of cycles where the CPU had an SME2 unit allocated to it, such that the SME and Streaming SVE state of the CPU is held in that SME2 to the total number of CPU cycles.

Units

This unit is expressed as percent of cycles.

Formula

CYCLES_CME_ALLOC / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
CYCLES_CME_ALLOC

Metric group

Cycle_Accounting

Methodology

Stage 2

cme_arb_pending_ratio, SME2 Arbitration Pending Cycles Ratio, metric

This metric is measures the ratio of cycles where the CPU is in arbitration while attempting to access an SME2 unit to the total number of CPU cycles.

Units

This unit is expressed as percent of cycles.

Formula

CYCLES_ARB_PENDING_CME / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
CYCLES_ARB_PENDING_CME

Metric group

Cycle_Accounting

Methodology

Stage 2

frontend_stalled_cycles, Frontend Stalled Cycles, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
STALL_FRONTEND

Metric group

Cycle_Accounting

Methodology

Stage 2

sm_active_cycles_ratio, SM Active Cycles Ratio, metric

This metric is measures the ratio of cycles when PSTATE.SM was enabled to the total number of CPU cycles.

Units

This unit is expressed as percent of cycles.

Formula

SM_ACTIVE_CYCLES / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
SM_ACTIVE_CYCLES

Metric group

Cycle_Accounting

Methodology

Stage 2

za_active_cycles_ratio, ZA Active Cycles Ratio, metric

This metric is measures the ratio of cycles when PSTATE.ZA was enabled to the total number of CPU cycles.

Units

This unit is expressed as percent of cycles.

Formula

ZA_ACTIVE / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
ZA_ACTIVE

Metric group

Cycle_Accounting

Methodology

Stage 2

5.5 Topdown_CME metrics for C1-Nano

Topdown SME2. This metric group contains a set of metrics to analyse an SME2 bound workload.

Summary of metrics in Topdown_CME:

- Total metrics: 3

Table 5-5: Topdown_CME metrics summary

Metric	Name	Description
backend_cme_backpressure_bound	Backend SME2 Backpressure Bound	This metric is the percentage of total cycles stalled in the backend as the SME2 unit is busy due...
backend_cme_busy_arb_bound	Backend SME2 Arbitration Bound	This metric is the percentage of total cycles stalled in the backend because an SME2 unit is...
backend_cme_cpu_bound	Backend SME2 CPU Bound	This metric is the percentage of total cycles stalled in the SME2 unit of the backend due to...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

backend_cme_backpressure_bound, Backend SME2 Backpressure Bound, metric

This metric is the percentage of total cycles stalled in the backend as the SME2 unit is busy due to other reasons.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_BUSY_CMEBOUND / STALL_BACKEND_BUSY_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_BUSY_CME

STALL_BACKEND_BUSY_CMEBOUND

Metric group

Topdown_CME

Methodology

Stage 1

backend_cme_busy_arb_bound, Backend SME2 Arbitration Bound, metric

This metric is the percentage of total cycles stalled in the backend because an SME2 unit is busy. The instruction cannot be sent to the SME2 unit because it is waiting for arbitration.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_BUSY_CME_ARB / STALL_BACKEND_BUSY_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_BUSY_CME
STALL_BACKEND_BUSY_CME_ARB

Metric group

Topdown_CME

Methodology

Stage 1

backend_cme_cpu_bound, Backend SME2 CPU Bound, metric

This metric is the percentage of total cycles stalled in the SME2 unit of the backend due to dependency to the other units of CPU for execution.

Units

This unit is expressed as percent of cycles.

Formula

STALL_BACKEND_BUSY_CME_CPUBOUND / STALL_BACKEND_BUSY_CME * 100

Related telemetry artifacts

Events

STALL_BACKEND_BUSY_CME
STALL_BACKEND_BUSY_CME_CPUBOUND

Metric group

Topdown_CME

Methodology

Stage 1

5.6 Topdown_L1 metrics for C1-Nano

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

Summary of metrics in Topdown_L1:

- Total metrics: 4

Table 5-6: Topdown_L1 metrics summary

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

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

backend_bound, Backend Bound, metric

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

Units

This unit is expressed as percent of slots.

Formula

STALL_SLOT_BACKEND / (3 * CPU_CYCLES) * 100

Related telemetry artifacts

Events

CPU_CYCLES
STALL_SLOT_BACKEND

Metric group

Topdown_L1

Methodology

Stage 1

bad_speculation, Bad Speculation, metric

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

Units

This unit is expressed as percent of slots.

Formula

(1 - STALL_SLOT / (3 * CPU_CYCLES)) * (1 - OP_RETIRED / OP_SPEC) * 100 + STALL_FRONTEND_FLUSH / CPU_CYCLES * 100

Related telemetry artifacts

Events

CPU_CYCLES
OP_RETIRED
OP_SPEC
STALL_FRONTEND_FLUSH

STALL_SLOT

Metric group

Topdown_L1

Methodology

Stage 1

frontend_bound, Frontend Bound, metric

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

Units

This unit is expressed as percent of slots.

Formula

(STALL_SLOT_FRONTEND / (3 * CPU_CYCLES) - STALL_FRONTEND_FLUSH / CPU_CYCLES) * 100

Related telemetry artifacts

Events

CPU_CYCLES
STALL_FRONTEND_FLUSH
STALL_SLOT_FRONTEND

Metric group

Topdown_L1

Methodology

Stage 1

retiring, Retiring, metric

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

Units

This unit is expressed as percent of slots.

Formula

(1 - STALL_SLOT / (CPU_CYCLES * 3)) * (OP_RETIRED / OP_SPEC) * 100

Related telemetry artifacts

Events

CPU_CYCLES
OP_RETIRED
OP_SPEC
STALL_SLOT

Metric group

Topdown_L1

Methodology

Stage 1

5.7 Topdown_Frontend metrics for C1-Nano

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

Summary of metrics in Topdown_Frontend:

- Total metrics: 8

Table 5-7: Topdown_Frontend metrics summary

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

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

frontend_cache_l1i_bound, Frontend Cache L1I Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_L1I / (STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) * 100

Related telemetry artifacts

Events

STALL_FRONTEND_L1I

STALL_FRONTEND_MEM

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_cache_l2i_bound, Frontend Cache L2I Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_MEM / (STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) * 100

Related telemetry artifacts

Events

STALL_FRONTEND_L1I
STALL_FRONTEND_MEM

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_core_bound, Frontend Core Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_CPUBOUND / STALL_FRONTEND * 100

Related telemetry artifacts

Events

STALL_FRONTEND
STALL_FRONTEND_CPUBOUND

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_core_flow_bound, Frontend Core Flow Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_FLOW / STALL_FRONTEND_CPUBOUND * 100

Related telemetry artifacts

Events

STALL_FRONTEND_CPUBOUND
STALL_FRONTEND_FLOW

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_core_flush_bound, Frontend Core Flush Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_FLUSH / STALL_FRONTEND_CPUBOUND * 100

Related telemetry artifacts

Events

STALL_FRONTEND_CPUBOUND
STALL_FRONTEND_FLUSH

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_mem_bound, Frontend Memory Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_MEMBOUND / STALL_FRONTEND * 100

Related telemetry artifacts

Events

STALL_FRONTEND
STALL_FRONTEND_MEMBOUND

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_mem_cache_bound, Frontend Mem Cache Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

(STALL_FRONTEND_L1I + STALL_FRONTEND_MEM) / STALL_FRONTEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_FRONTEND_L1I
STALL_FRONTEND_MEM
STALL_FRONTEND_MEMBOUND

Metric group

Topdown_Frontend

Methodology

Stage 1

frontend_mem_tlb_bound, Frontend Mem TLB Bound, metric

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

Units

This unit is expressed as percent of cycles.

Formula

STALL_FRONTEND_TLB / STALL_FRONTEND_MEMBOUND * 100

Related telemetry artifacts

Events

STALL_FRONTEND_MEMBOUND
STALL_FRONTEND_TLB

Metric group

Topdown_Frontend

Methodology

Stage 1

5.8 General metrics for C1-Nano

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

Summary of metrics in General:

- Total metrics: 1

Table 5-8: General metrics summary

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

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

ipc, Instructions Per Cycle, metric

This metric measures the number of instructions retired per cycle.

Units

This unit is expressed as per cycle.

Formula

INST_RETIRED / CPU_CYCLES

Related telemetry artifacts

Events

CPU_CYCLES
INST_RETIRED

Metric group

General

Methodology

Stage 2

5.9 MPKI metrics for C1-Nano

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: 12

Table 5-9: MPKI metrics summary

Metric	Name	Description
branch_mpki	Branch MPKI	This metric measures the number of branch mispredictions per thousand instructions executed.
dtlb_mpki	DTLB MPKI	This metric measures the number of data TLB Walks per thousand instructions executed.
itlb_mpki	ITLB MPKI	This metric measures the number of instruction TLB Walks per thousand instructions executed.
l1d_cache_mpki	L1D Cache MPKI	This metric measures the number of level 1 data cache accesses missed per thousand instructions...
l1d_tlb_mpki	L1 Data TLB MPKI	This metric measures the number of level 1 data TLB accesses missed per thousand instructions...
l1i_cache_mpki	L1I Cache MPKI	This metric measures the number of level 1 instruction cache accesses missed per thousand...
l1i_tlb_mpki	L1 Instruction TLB MPKI	This metric measures the number of level 1 instruction TLB accesses missed per thousand...
l2_tlb_mpki	L2 Unified TLB MPKI	This metric measures the number of level 2 unified TLB accesses missed per thousand instructions...
l2d_cache_mpki	L2D Cache MPKI	This metric measures the number of level 2 unified cache data accesses missed per thousand...
l2i_cache_mpki	L2I Cache MPKI	This metric measures the number of level 2 unified cache instruction accesses missed per thousand...
l3_cache_mpki	L3 Cache MPKI	This metric measures the number of level 3 unified cache accesses missed per thousand...
ll_cache_read_mpki	LL Cache Read MPKI	This metric measures the number of last level cache read accesses missed per thousand...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

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

BR_MIS_PRED_RETIRED / 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

DTLB_WALK / 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

ITLB_WALK / 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

(L1D_CACHE_REFILL_RD + L1D_CACHE_REFILL_WR) / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED
L1D_CACHE_REFILL_RD
L1D_CACHE_REFILL_WR

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

L1D_TLB_REFILL / 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

L1I_CACHE_REFILL / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED
L1I_CACHE_REFILL

Metric group

MPKI

Other metric group: L1I_Cache_Effectiveness

Methodology

Stage 2

l1i_tlb_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

L1I_TLB_REFILL / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

L1I_TLB_REFILL

Metric group

MPKI

Other metric group: ITLB_Effectiveness

Methodology

Stage 2

l2_tlb_mpki, L2 Unified TLB MPKI, metric

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

Units

This unit is expressed as mpki.

Formula

L2D_TLB_REFILL / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

L2D_TLB_REFILL

Metric group

MPKI

Other metric group: DTLB_Effectiveness

Other metric group: ITLB_Effectiveness

Methodology

Stage 2

l2d_cache_mpki, L2D Cache MPKI, metric

This metric measures the number of level 2 unified cache data 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

(L2D_CACHE_REFILL_RD + L2D_CACHE_REFILL_WR) / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED
L2D_CACHE_REFILL_RD
L2D_CACHE_REFILL_WR

Metric group

MPKI

Other metric group: L2D_Cache_Effectiveness

Methodology

Stage 2

l2i_cache_mpki, L2I Cache MPKI, metric

This metric measures the number of level 2 unified cache instruction 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

L2I_CACHE_REFILL / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED
L2I_CACHE_REFILL

Metric group

MPKI

Other metric group: L2I_Cache_Effectiveness

Methodology

Stage 2

l3_cache_mpki, L3 Cache MPKI, metric

This metric measures the number of level 3 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

L3D_CACHE_REFILL_RD / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

L3D_CACHE_REFILL_RD

Metric group

MPKI

Other metric group: L3_Cache_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

LL_CACHE_MISS_RD / INST_RETIRED * 1000

Related telemetry artifacts

Events

INST_RETIRED

LL_CACHE_MISS_RD

Metric group

MPKI

Other metric group: LL_Cache_Effectiveness

Methodology

Stage 2

5.10 Miss_Ratio metrics for C1-Nano

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

Summary of metrics in Miss_Ratio:

- Total metrics: 12

Table 5-10: Miss_Ratio metrics summary

Metric	Name	Description
branch_misprediction_ratio	Branch Misprediction Ratio	This metric measures the ratio of branches mispredicted to the total number of branches...
dtlb_walk_ratio	DTLB Walk Ratio	This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This...
itlb_walk_ratio	ITLB Walk Ratio	This metric measures the ratio of instruction TLB Walks to the total number of instruction TLB...
l1d_cache_miss_ratio	L1D Cache Miss Ratio	This metric measures the ratio of level 1 data cache accesses missed to the total number of level...
l1d_tlb_miss_ratio	L1 Data TLB Miss Ratio	This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1...
l1i_cache_miss_ratio	L1I Cache Miss Ratio	This metric measures the ratio of level 1 instruction cache accesses missed to the total number...
l1i_tlb_miss_ratio	L1 Instruction TLB Miss Ratio	This metric measures the ratio of level 1 instruction TLB accesses missed to the total number of...
l2_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...
l2d_cache_miss_ratio	L2D Cache Miss Ratio	This metric measures the ratio of level 2 cache data accesses missed to the total number of level...
l2i_cache_miss_ratio	L2I Cache Miss Ratio	This metric measures the ratio of level 2 cache instruction accesses missed to the total number...
l3_cache_miss_ratio	L3 Cache Miss Ratio	This metric measures the ratio of level 3 cache accesses missed to the total number of level 3...
ll_cache_read_miss_ratio	LL Cache Read Miss Ratio	This metric measures the ratio of last level cache read accesses missed to the total number of...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

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

DTLB_WALK / L1D_TLB

Related telemetry artifacts

Events

DTLB_WALK
L1D_TLB

Metric group

Miss_Ratio

Other metric group: DTLB_Effectiveness

Methodology

Stage 2

itlb_walk_ratio, ITLB Walk Ratio, metric

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

Units

This unit is expressed as per tlb access.

Formula

ITLB_WALK / 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

(L1D_CACHE_REFILL_RD + L1D_CACHE_REFILL_WR) / L1D_CACHE_RW

Related telemetry artifacts

Events

L1D_CACHE_REFILL_RD

L1D_CACHE_REFILL_WR

L1D_CACHE_RW

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

L1D_TLB_REFILL / 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

L1I_CACHE_REFILL / L1I_CACHE

Related telemetry artifacts

Events

L1I_CACHE
L1I_CACHE_REFILL

Metric group

Miss_Ratio

Other metric group: L1I_Cache_Effectiveness

Methodology

Stage 2

l1i_tlb_miss_ratio, L1 Instruction TLB Miss Ratio, metric

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

Units

This unit is expressed as per tlb access.

Formula

L1I_TLB_REFILL / L1I_TLB

Related telemetry artifacts

Events

L1I_TLB
L1I_TLB_REFILL

Metric group

Miss_Ratio

Other metric group: ITLB_Effectiveness

Methodology

Stage 2

l2_tlb_miss_ratio, L2 Unified TLB Miss Ratio, metric

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

Units

This unit is expressed as per tlb access.

Formula

L2D_TLB_REFILL / L2D_TLB

Related telemetry artifacts

Events

L2D_TLB
L2D_TLB_REFILL

Metric group

Miss_Ratio
Other metric group: DTLB_Effectiveness
Other metric group: ITLB_Effectiveness

Methodology

Stage 2

l2d_cache_miss_ratio, L2D Cache Miss Ratio, metric

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

Units

This unit is expressed as per cache access.

Formula

(L2D_CACHE_REFILL_RD + L2D_CACHE_REFILL_WR) / L2D_CACHE_RW

Related telemetry artifacts

Events

L2D_CACHE_REFILL_RD
L2D_CACHE_REFILL_WR
L2D_CACHE_RW

Metric group

Miss_Ratio
Other metric group: L2D_Cache_Effectiveness

Methodology

Stage 2

l2i_cache_miss_ratio, L2I Cache Miss Ratio, metric

This metric measures the ratio of level 2 cache instruction accesses missed to the total number of level 2 cache instruction accesses. This gives an indication of the effectiveness of instruction accesses in 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

L2I_CACHE_REFILL / L2I_CACHE

Related telemetry artifacts

Events

L2I_CACHE
L2I_CACHE_REFILL

Metric group

Miss_Ratio

Other metric group: L2I_Cache_Effectiveness

Methodology

Stage 2

l3_cache_miss_ratio, L3 Cache Miss Ratio, metric

This metric measures the ratio of level 3 cache accesses missed to the total number of level 3 cache accesses. This gives an indication of the effectiveness of the level 3 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

L3D_CACHE_REFILL_RD / L3D_CACHE_RD

Related telemetry artifacts

Events

L3D_CACHE_RD
L3D_CACHE_REFILL_RD

Metric group

Miss_Ratio

Other metric group: L3_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

LL_CACHE_MISS_RD / LL_CACHE_RD

Related telemetry artifacts

Events

LL_CACHE_MISS_RD
LL_CACHE_RD

Metric group

Miss_Ratio
Other metric group: LL_Cache_Effectiveness

Methodology

Stage 2

5.11 SVE_Effectiveness metrics for C1-Nano

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

Summary of metrics in SVE_Effectiveness:

- Total metrics: 4

Table 5-11: SVE_Effectiveness metrics summary

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

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

sve_predicate_empty_percentage, SVE Empty Predicate Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

SVE_PRED_EMPTY_SPEC / SVE_PRED_SPEC * 100

Related telemetry artifacts

Events

SVE_PRED_EMPTY_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

sve_predicate_full_percentage, SVE Full Predicate Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

SVE_PRED_FULL_SPEC / SVE_PRED_SPEC * 100

Related telemetry artifacts

Events

SVE_PRED_FULL_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

sve_predicate_partial_percentage, SVE Partial Predicate Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

SVE_PRED_PARTIAL_SPEC / SVE_PRED_SPEC * 100

Related telemetry artifacts

Events

SVE_PRED_PARTIAL_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

sve_predicate_percentage, SVE Predicate Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

SVE_PRED_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC
SVE_PRED_SPEC

Metric group

SVE_Effectiveness

Methodology

Stage 2

5.12 FP_Precision_Mix metrics for C1-Nano

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

Summary of metrics in FP_Precision_Mix:

- Total metrics: 3

Table 5-12: FP_Precision_Mix metrics summary

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

Metric	Name	Description
fp64_percentage	Double Precision Floating Point Percentage	This metric measures double-precision floating point operations as a percentage of operations...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

fp16_percentage, Half Precision Floating Point Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

FP_HP_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

FP_HP_SPEC
INST_SPEC

Metric group

FP_Precision_Mix

Methodology

Stage 2

fp32_percentage, Single Precision Floating Point Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

FP_SP_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

FP_SP_SPEC
INST_SPEC

Metric group

FP_Precision_Mix

Methodology

Stage 2

fp64_percentage, Double Precision Floating Point Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

FP_DP_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

FP_DP_SPEC
INST_SPEC

Metric group

FP_Precision_Mix

Methodology

Stage 2

5.13 Branch_Effectiveness metrics for C1-Nano

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

Summary of metrics in Branch_Effectiveness:

- Total metrics: 5

Table 5-13: Branch_Effectiveness metrics summary

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

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

branch_direct_ratio, Branch Direct Ratio, metric

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

Units

This unit is expressed as per branch.

Formula

BR_IMMED_RETIRED / BR_RETIRED

Related telemetry artifacts

Events

BR_IMMED_RETIRED
BR_RETIRED

Metric group

Branch_Effectiveness

Methodology

Stage 2

branch_indirect_ratio, Branch Indirect Ratio, metric

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

Units

This unit is expressed as per branch.

Formula

BR_IND_RETIRED / BR_RETIRED

Related telemetry artifacts

Events

BR_IND_RETIRED
BR_RETIRED

Metric group

Branch_Effectiveness

Methodology

Stage 2

branch_misprediction_ratio**, Branch Misprediction Ratio, metric

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

Units

This unit is expressed as per branch.

Formula

BR_MIS_PRED_RETIRED / 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

BR_MIS_PRED_RETIRED / 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

branch_return_ratio, Branch Return Ratio, metric

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

Units

This unit is expressed as per branch.

Formula

BR_RETURN_RETIRED / BR_RETIRED

Related telemetry artifacts

Events

BR_RETIRED

BR_RETURN_RETIRED

Metric group

Branch_Effectiveness

Methodology

Stage 2

5.14 ITLB_Effectiveness metrics for C1-Nano

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: 10

Table 5-14: 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_average_depth	ITLB Walk Average Depth of Accesses	This metric measures the average depth of the Instruction TLB walks for an instruction TLB refill
itlb_walk_average_latency	ITLB Walk Average Latency	This metric measures the average latency of instruction TLB walks in CPU cycles
itlb_walk_block_ratio	ITLB Walk Block Ratio	This metric measures the ratio of instruction TLB Walks that returned a block to the total number...
itlb_walk_page_ratio	ITLB Walk Page Ratio	This metric measures the ratio of instruction TLB Walks that returned a page to the total number...
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 C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

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

ITLB_WALK / INST_RETIRED * 1000

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

Related telemetry artifacts

Events

INST_RETIRED

ITLB_WALK

Metric group

ITLB_Effectiveness

Other metric group: MPKI

Methodology

Stage 2

itlb_walk_average_depth, ITLB Walk Average Depth of Accesses, metric

This metric measures the average depth of the Instruction TLB walks for an instruction TLB refill

Units

This unit is expressed as page accesses.

Formula

ITLB_STEP / ITLB_WALK

Related telemetry artifacts

Events

ITLB_STEP

ITLB_WALK

Metric group

ITLB_Effectiveness

Methodology

Stage 2

itlb_walk_average_latency, ITLB Walk Average Latency, metric

This metric measures the average latency of instruction TLB walks in CPU cycles

Units

This unit is expressed in cycles..

Formula

ITLB_WALK_PERCYC / ITLB_WALK

Related telemetry artifacts

Events

ITLB_WALK

ITLB_WALK_PERCYC

Metric group

ITLB_Effectiveness

Other metric group: Average_Latency

Methodology

Stage 2

itlb_walk_block_ratio, ITLB Walk Block Ratio, metric

This metric measures the ratio of instruction TLB Walks that returned a block to the total number of instruction TLB accesses. Block size is any memory block larger than the page granule size set by the system.

Units

This unit is expressed as per tlb access.

Formula

ITLB_WALK_BLOCK / L1I_TLB

Related telemetry artifacts

Events

ITLB_WALK_BLOCK

L1I_TLB

Metric group

ITLB_Effectiveness

Methodology

Stage 2

itlb_walk_page_ratio, ITLB Walk Page Ratio, metric

This metric measures the ratio of instruction TLB Walks that returned a page to the total number of instruction TLB accesses. Page size is determined by the page granule size set by the system.

Units

This unit is expressed as per tlb access.

Formula

ITLB_WALK_PAGE / L1I_TLB

Related telemetry artifacts

Events

ITLB_WALK_PAGE

L1I_TLB

Metric group

ITLB_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

ITLB_WALK / 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

L1I_TLB_REFILL / 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_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

L1I_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
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_mpki**, L2 Unified TLB MPKI, metric

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

Units

This unit is expressed as mpki.

Formula

L2D_TLB_REFILL / INST_RETIRED * 1000

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

Related telemetry artifacts

Events

INST_RETIRED
L2D_TLB_REFILL

Metric group

ITLB_Effectiveness
Other metric group: DTLB_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

5.15 DTLB_Effectiveness metrics for C1-Nano

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: 10

Table 5-15: DTLB_Effectiveness metrics summary

Metric	Name	Description
dtlb_mpki	DTLB MPKI	This metric measures the number of data TLB Walks per thousand instructions executed.
dtlb_walk_average_depth	DTLB Walk Average Depth of Accesses	This metric measures the average depth of the Instruction TLB walks for a data TLB refill
dtlb_walk_average_latency	DTLB Walk Average Latency	This metric measures the average latency of data TLB walks in CPU cycles
dtlb_walk_block_ratio	DTLB Walk Block Ratio	This metric measures the ratio of data TLB Walks that returned a block to the total number of...
dtlb_walk_page_ratio	DTLB Walk Page Ratio	This metric measures the ratio of data TLB Walks that returned a page to the total number of data...
dtlb_walk_ratio	DTLB Walk Ratio	This metric measures the ratio of data TLB Walks to the total number of data TLB accesses. This...

Metric	Name	Description
l1d_tlb_miss_ratio	L1 Data TLB Miss Ratio	This metric measures the ratio of level 1 data TLB accesses missed to the total number of level 1...
l1d_tlb_mpki	L1 Data TLB MPKI	This metric measures the number of level 1 data TLB accesses missed per thousand instructions...
l2_tlb_miss_ratio	L2 Unified TLB Miss Ratio	This metric measures the ratio of level 2 unified TLB accesses missed to the total number of...
l2_tlb_mpki	L2 Unified TLB MPKI	This metric measures the number of level 2 unified TLB accesses missed per thousand instructions...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

dtlb_mpki**, DTLB MPKI, metric

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

Units

This unit is expressed as mpki.

Formula

DTLB_WALK / INST_RETIRED * 1000

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

Related telemetry artifacts

Events

DTLB_WALK
INST_RETIRED

Metric group

DTLB_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

dtlb_walk_average_depth, DTLB Walk Average Depth of Accesses, metric

This metric measures the average depth of the Instruction TLB walks for a data TLB refill

Units

This unit is expressed as page table accesses.

Formula

DTLB_STEP / DTLB_WALK

Related telemetry artifacts

Events

DTLB_STEP
DTLB_WALK

Metric group

DTLB_Effectiveness

Methodology

Stage 2

dtlb_walk_average_latency, DTLB Walk Average Latency, metric

This metric measures the average latency of data TLB walks in CPU cycles

Units

This unit is expressed in cycles..

Formula

DTLB_WALK_PERCYC / DTLB_WALK

Related telemetry artifacts

Events

DTLB_WALK

DTLB_WALK_PERCYC

Metric group

DTLB_Effectiveness

Other metric group: Average_Latency

Methodology

Stage 2

dtlb_walk_block_ratio, DTLB Walk Block Ratio, metric

This metric measures the ratio of data TLB Walks that returned a block to the total number of data TLB accesses. Block size is any memory block larger than the page granule size set by the system.

Units

This unit is expressed as per tlb access.

Formula

DTLB_WALK_BLOCK / L1D_TLB

Related telemetry artifacts

Events

DTLB_WALK_BLOCK

L1D_TLB

Metric group

DTLB_Effectiveness

Methodology

Stage 2

dtlb_walk_page_ratio, DTLB Walk Page Ratio, metric

This metric measures the ratio of data TLB Walks that returned a page to the total number of data TLB accesses. Page size is determined by the page granule size set by the system.

Units

This unit is expressed as per tlb access.

Formula

DTLB_WALK_PAGE / L1D_TLB

Related telemetry artifacts

Events

DTLB_WALK_PAGE

L1D_TLB

Metric group

DTLB_Effectiveness

Methodology

Stage 2

dtlb_walk_ratio**, DTLB Walk Ratio, metric

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

Units

This unit is expressed as per tlb access.

Formula

DTLB_WALK / L1D_TLB

** 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_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

L1D_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

L1D_TLB_REFILL

Metric group

DTLB_Effectiveness

Other metric group: MPKI

Methodology

Stage 2

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

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

Units

This unit is expressed as per tlb access.

Formula

L2D_TLB_REFILL / L2D_TLB

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

Related telemetry artifacts

Events

L2D_TLB

L2D_TLB_REFILL

Metric group

DTLB_Effectiveness

Other metric group: ITLB_Effectiveness

Other metric group: Miss_Ratio

Methodology

Stage 2

l2_tlb_mpki***, L2 Unified TLB MPKI, metric

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

Units

This unit is expressed as mpki.

Formula

L2D_TLB_REFILL / INST_RETIRED * 1000

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

Related telemetry artifacts

Events

INST_RETIRED

L2D_TLB_REFILL

Metric group

DTLB_Effectiveness

Other metric group: ITLB_Effectiveness

Other metric group: MPKI

Methodology

Stage 2

5.16 L1I_Cache_Effectiveness metrics for C1-Nano

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

Summary of metrics in L1I_Cache_Effectiveness:

- Total metrics: 2

Table 5-16: L1I_Cache_Effectiveness metrics summary

Metric	Name	Description
l1i_cache_miss_ratio	L1I Cache Miss Ratio	This metric measures the ratio of level 1 instruction cache accesses missed to the total number...
l1i_cache_mpki	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 C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

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

L1I_CACHE_REFILL / L1I_CACHE

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

Related telemetry artifacts

Events

L1I_CACHE
L1I_CACHE_REFILL

Metric group

L1I_Cache_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l1i_cache_mpki**, L1I Cache MPKI, metric

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

Units

This unit is expressed as mpki.

Formula

L1I_CACHE_REFILL / INST_RETIRED * 1000

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

Related telemetry artifacts

Events

INST_RETIRED

L1I_CACHE_REFILL

Metric group

L1I_Cache_Effectiveness

Other metric group: MPKI

Methodology

Stage 2

5.17 L1D_Cache_Effectiveness metrics for C1-Nano

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

Summary of metrics in L1D_Cache_Effectiveness:

- Total metrics: 2

Table 5-17: L1D_Cache_Effectiveness metrics summary

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

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

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_RD + L1D_CACHE_REFILL_WR) / L1D_CACHE_RW

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

Related telemetry artifacts

Events

L1D_CACHE_REFILL_RD

L1D_CACHE_REFILL_WR

L1D_CACHE_RW

Metric group

L1D_Cache_Effectiveness

Other metric group: Miss_Ratio

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

(L1D_CACHE_REFILL_RD + L1D_CACHE_REFILL_WR) / 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_RD
L1D_CACHE_REFILL_WR

Metric group

L1D_Cache_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

5.18 L2I_Cache_Effectiveness metrics for C1-Nano

L2I Unified Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of instruction access in the L2 Unified Cache on this processor.

Summary of metrics in L2I_Cache_Effectiveness:

- Total metrics: 2

Table 5-18: L2I_Cache_Effectiveness metrics summary

Metric	Name	Description
l2i_cache_miss_ratio	L2I Cache Miss Ratio	This metric measures the ratio of level 2 cache instruction accesses missed to the total number...
l2i_cache_mpki	L2I Cache MPKI	This metric measures the number of level 2 unified cache instruction accesses missed per thousand...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

l2i_cache_miss_ratio**, L2I Cache Miss Ratio, metric

This metric measures the ratio of level 2 cache instruction accesses missed to the total number of level 2 cache instruction accesses. This gives an indication of the effectiveness of instruction accesses in 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

L2I_CACHE_REFILL / L2I_CACHE

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

Related telemetry artifacts

Events

L2I_CACHE
L2I_CACHE_REFILL

Metric group

L2I_Cache_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l2i_cache_mpki**, L2I Cache MPKI, metric

This metric measures the number of level 2 unified cache instruction 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

L2I_CACHE_REFILL / INST_RETIRED * 1000

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

Related telemetry artifacts

Events

INST_RETIRED
L2I_CACHE_REFILL

Metric group

L2I_Cache_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

5.19 L2D_Cache_Effectiveness metrics for C1-Nano

L2D Unified Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of data access in the L2 Unified Cache on this processor.

Summary of metrics in L2D_Cache_Effectiveness:

- Total metrics: 2

Table 5-19: L2D_Cache_Effectiveness metrics summary

Metric	Name	Description
l2d_cache_miss_ratio	L2D Cache Miss Ratio	This metric measures the ratio of level 2 cache data accesses missed to the total number of level...
l2d_cache_mpki	L2D Cache MPKI	This metric measures the number of level 2 unified cache data accesses missed per thousand...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

l2d_cache_miss_ratio**, L2D Cache Miss Ratio, metric

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

Units

This unit is expressed as per cache access.

Formula

(L2D_CACHE_REFILL_RD + L2D_CACHE_REFILL_WR) / L2D_CACHE_RW

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

Related telemetry artifacts

Events

L2D_CACHE_REFILL_RD
L2D_CACHE_REFILL_WR
L2D_CACHE_RW

Metric group

L2D_Cache_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l2d_cache_mpki**, L2D Cache MPKI, metric

This metric measures the number of level 2 unified cache data 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

(L2D_CACHE_REFILL_RD + L2D_CACHE_REFILL_WR) / 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_RD
L2D_CACHE_REFILL_WR

Metric group

L2D_Cache_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

5.20 L3_Cache_Effectiveness metrics for C1-Nano

L3 Unified Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of L3 Unified Cache on this processor.

Summary of metrics in L3_Cache_Effectiveness:

- Total metrics: 2

Table 5-20: L3_Cache_Effectiveness metrics summary

Metric	Name	Description
l3_cache_miss_ratio	L3 Cache Miss Ratio	This metric measures the ratio of level 3 cache accesses missed to the total number of level 3...
l3_cache_mpki	L3 Cache MPKI	This metric measures the number of level 3 unified cache accesses missed per thousand...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

l3_cache_miss_ratio**, L3 Cache Miss Ratio, metric

This metric measures the ratio of level 3 cache accesses missed to the total number of level 3 cache accesses. This gives an indication of the effectiveness of the level 3 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

L3D_CACHE_REFILL_RD / L3D_CACHE_RD

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

Related telemetry artifacts

Events

L3D_CACHE_RD
L3D_CACHE_REFILL_RD

Metric group

L3_Cache_Effectiveness
Other metric group: Miss_Ratio

Methodology

Stage 2

l3_cache_mpki**, L3 Cache MPKI, metric

This metric measures the number of level 3 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

L3D_CACHE_REFILL_RD / INST_RETIRED * 1000

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

Related telemetry artifacts

Events

INST_RETIRED
L3D_CACHE_REFILL_RD

Metric group

L3_Cache_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

5.21 LL_Cache_Effectiveness metrics for C1-Nano

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

Summary of metrics in LL_Cache_Effectiveness:

- Total metrics: 3

Table 5-21: LL_Cache_Effectiveness metrics summary

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

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

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

(LL_CACHE_RD - LL_CACHE_MISS_RD) / 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

LL_CACHE_MISS_RD / 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_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

LL_CACHE_MISS_RD / INST_RETIRED * 1000

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

Related telemetry artifacts

Events

INST_RETIRED
LL_CACHE_MISS_RD

Metric group

LL_Cache_Effectiveness
Other metric group: MPKI

Methodology

Stage 2

5.22 Operation_Mix metrics for C1-Nano

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: 12

Table 5-22: Operation_Mix metrics summary

Metric	Name	Description
branch_percentage	Branch Operations Percentage	This metric measures branch operations as a percentage of operations speculatively executed.
crypto_percentage	Crypto Operations Percentage	This metric measures crypto operations as a percentage of operations speculatively executed.
integer_dp_percentage	Integer Operations Percentage	This metric measures scalar integer operations as a percentage of operations speculatively executed.
load_ls_percentage	Load Operations Percentage of Load/ Store Operations	This metric measures load operations as a percentage of load and store operations speculatively...
load_percentage	Load Operations Percentage	This metric measures load operations as a percentage of operations speculatively executed.
load_store_percentage	Load/Store Operations Percentage	This metric measures load and store operations as a percentage of operations speculatively executed.
scalar_fp_percentage	Floating Point Operations Percentage	This metric measures scalar floating point operations as a percentage of operations speculatively...
simd_percentage	Advanced SIMD Operations Percentage	This metric measures advanced SIMD operations as a percentage of total operations speculatively...
sme_percentage	SME Operations Percentage	This metric measures Scalable Matrix extension data processing operations as a percentage of...
store_ls_percentage	Store Operations Percentage of Load/ Store Operations	This metric measures store operations as a percentage of load and store operations speculatively...
store_percentage	Store Operations Percentage	This metric measures store operations as a percentage of operations speculatively executed.
sve_percentage	Scalable Vector Operations Percentage	This metric measures scalable vector operations as a percentage of operations speculatively...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

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

(BR_IMMED_SPEC + BR_INDIRECT_SPEC) / 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

CRYPTO_SPEC / 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

DP_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

DP_SPEC
INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

load_ls_percentage, Load Operations Percentage of Load/Store Operations, metric

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

Units

This unit is expressed as percent of load/store operations.

Formula

LD_SPEC / LDST_SPEC * 100

Related telemetry artifacts

Events

LDST_SPEC
LD_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

LD_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC
LD_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

load_store_percentage, Load/Store Operations Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

LDST_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC

LDST_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

VFP_SPEC / 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

ASE_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

ASE_SPEC

INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

sme_percentage, SME Operations Percentage, metric

This metric measures Scalable Matrix extension data processing operations as a percentage of operations speculatively executed.

Units

This unit is expressed as percent of operations.

Formula

SME_INST_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC
SME_INST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

store_ls_percentage, Store Operations Percentage of Load/Store Operations, metric

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

Units

This unit is expressed as percent of load/store operations.

Formula

ST_SPEC / LDST_SPEC * 100

Related telemetry artifacts

Events

LDST_SPEC
ST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

store_percentage, Store Operations Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

ST_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC

ST_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

sve_percentage, Scalable Vector Operations Percentage, metric

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

Units

This unit is expressed as percent of operations.

Formula

SVE_SPEC / INST_SPEC * 100

Related telemetry artifacts

Events

INST_SPEC

SVE_SPEC

Metric group

Operation_Mix

Methodology

Stage 2

5.23 Prefetcher_Effectiveness metrics for C1-Nano

Prefetcher Effectiveness. L2 prefetcher effectiveness metrics.

Summary of metrics in Prefetcher_Effectiveness:

- Total metrics: 3

Table 5-23: Prefetcher_Effectiveness metrics summary

Metric	Name	Description
l2_prefetcher_accuracy_l1hwprf_exclusive	L2 Cache Prefetcher Accuracy (L1 HW prefetch exclusive)	This metric measures L2D cache prefetcher accuracy excluding L1 hardware prefetch requests
l2_prefetcher_coverage_l1hwprf_exclusive	L2 Cache Prefetcher Coverage (L1 HW prefetch exclusive)	This metric measures L2D cache prefetcher coverage excluding L1 hardware prefetch requests
l2_prefetcher_timeliness_l1hwprf_exclusive	L2 Cache Prefetcher Timeliness (L1 HW prefetch exclusive)	This metric measures L2D cache prefetcher timeliness excluding L1 hardware prefetch requests

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

l2_prefetcher_accuracy_l1hwprf_exclusive, L2 Cache Prefetcher Accuracy (L1 HW prefetch exclusive), metric

This metric measures L2D cache prefetcher accuracy excluding L1 hardware prefetch requests

Units

This unit is expressed as useful prefetches per total prefetches.

Formula

(L2D_CACHE_HIT_RW_FHWPRF + L2D_LFB_HIT_RW_FHWPRF) / L2D_CACHE_REFILL_HWPRF

Related telemetry artifacts

Events

L2D_CACHE_HIT_RW_FHWPRF
L2D_CACHE_REFILL_HWPRF
L2D_LFB_HIT_RW_FHWPRF

Metric group

Prefetcher_Effectiveness

Methodology

Stage 2

l2_prefetcher_coverage_l1hwprf_exclusive, L2 Cache Prefetcher Coverage (L1 HW prefetch exclusive), metric

This metric measures L2D cache prefetcher coverage excluding L1 hardware prefetch requests

Units

This unit is expressed as useful prefetches per demand misses.

Formula

(L2D_CACHE_HIT_RW_FHWPRF + L2D_LFB_HIT_RW_FHWPRF) / (L2D_CACHE_HIT_RW_FHWPRF + L2D_LFB_HIT_RW_FHWPRF + L2D_CACHE_REFILL_RD + L2D_CACHE_REFILL_WR)

Related telemetry artifacts

Events

L2D_CACHE_HIT_RW_FHWPRF

L2D_CACHE_REFILL_RD
L2D_CACHE_REFILL_WR
L2D_LFB_HIT_RW_FHWPRF

Metric group

Prefetcher_Effectiveness

Methodology

Stage 2

l2_prefetcher_timeliness_l1hwprf_exclusive, L2 Cache Prefetcher Timeliness (L1 HW prefetch exclusive), metric

This metric measures L2D cache prefetcher timeliness excluding L1 hardware prefetch requests

Units

This unit is expressed as timely prefetches per useful prefetches.

Formula

L2D_CACHE_HIT_RW_FHWPRF / (L2D_CACHE_HIT_RW_FHWPRF + L2D_LFB_HIT_RW_FHWPRF)

Related telemetry artifacts

Events

L2D_CACHE_HIT_RW_FHWPRF
L2D_LFB_HIT_RW_FHWPRF

Metric group

Prefetcher_Effectiveness

Methodology

Stage 2

5.24 Average_Latency metrics for C1-Nano

Average Latency. This metric group contains metrics that provide average latencies of costly memory related operations by the CPU that can have variable number of cycles

Summary of metrics in Average_Latency:

- Total metrics: 5

Table 5-24: Average_Latency metrics summary

Metric	Name	Description
bus_read_requests_average_latency	Bus Read Request Average Latency	This metric measures the average latency of read requests made by this processor on the system...
dtlb_walk_average_latency	DTLB Walk Average Latency	This metric measures the average latency of data TLB walks in CPU cycles

Metric	Name	Description
instruction_fetch_average_latency	Instruction Fetch Average Latency	This metric measures the average latency of instruction fetches in CPU cycles
itlb_walk_average_latency	ITLB Walk Average Latency	This metric measures the average latency of instruction TLB walks in CPU cycles
load_average_latency	Load Operation Average Latency	This metric measures the average latency of load operations in CPU cycles

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

bus_read_requests_average_latency, Bus Read Request Average Latency, metric

This metric measures the average latency of read requests made by this processor on the system bus in CPU cycles

Units

This unit is expressed in cycles..

Formula

BUS_REQ_RD_PERCYC / BUS_REQ_RD

Related telemetry artifacts

Events

BUS_REQ_RD
BUS_REQ_RD_PERCYC

Metric group

Average_Latency
Other metric group: Bus_Effectiveness

Methodology

Stage 2

dtlb_walk_average_latency**, DTLB Walk Average Latency, metric

This metric measures the average latency of data TLB walks in CPU cycles

Units

This unit is expressed in cycles..

Formula

DTLB_WALK_PERCYC / DTLB_WALK

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

Related telemetry artifacts

Events

DTLB_WALK
DTLB_WALK_PERCYC

Metric group

Average_Latency

Other metric group: DTLB_Effectiveness

Methodology

Stage 2

instruction_fetch_average_latency, Instruction Fetch Average Latency, metric

This metric measures the average latency of instruction fetches in CPU cycles

Units

This unit is expressed in cycles..

Formula

INST_FETCH_PERCYC / INST_FETCH

Related telemetry artifacts

Events

INST_FETCH

INST_FETCH_PERCYC

Metric group

Average_Latency

Methodology

Stage 2

itlb_walk_average_latency**, ITLB Walk Average Latency, metric

This metric measures the average latency of instruction TLB walks in CPU cycles

Units

This unit is expressed in cycles..

Formula

ITLB_WALK_PERCYC / ITLB_WALK

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

Related telemetry artifacts

Events

ITLB_WALK

ITLB_WALK_PERCYC

Metric group

Average_Latency

Other metric group: ITLB_Effectiveness

Methodology

Stage 2

load_average_latency, Load Operation Average Latency, metric

This metric measures the average latency of load operations in CPU cycles

Units

This unit is expressed in cycles..

Formula

MEM_ACCESS_RD_PERCYC / MEM_ACCESS

Related telemetry artifacts

Events

MEM_ACCESS
MEM_ACCESS_RD_PERCYC

Metric group

Average_Latency

Methodology

Stage 2

5.25 Bus_Effectiveness metrics for C1-Nano

Bus Effectiveness. This metric group contains metrics to evaluate the effectiveness of bus transactions issued by this processor

Summary of metrics in Bus_Effectiveness:

- Total metrics: 1

Table 5-25: Bus_Effectiveness metrics summary

Metric	Name	Description
bus_read_requests_average_latency	Bus Read Request Average Latency	This metric measures the average latency of read requests made by this processor on the system...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

bus_read_requests_average_latency**, Bus Read Request Average Latency, metric

This metric measures the average latency of read requests made by this processor on the system bus in CPU cycles

Units

This unit is expressed in cycles..

Formula

BUS_REQ_RD_PERCYC / BUS_REQ_RD

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

Related telemetry artifacts

Events

BUS_REQ_RD
BUS_REQ_RD_PERCYC

Metric group

Bus_Effectiveness
Other metric group: Average_Latency

Methodology

Stage 2

5.26 System_Memory_Effectiveness metrics for C1-Nano

System Memory Effectiveness. This metric group contains a set of metrics to analyze a system memory bound workload, providing hierarchical data hits in system memory resources internal or external to the processor.

Summary of metrics in System_Memory_Effectiveness:

- Total metrics: 2

Table 5-26: System_Memory_Effectiveness metrics summary

Metric	Name	Description
system_l3_cache_hit_ratio	System L3 Cache Hit Ratio	This metric measures the ratio of system L3 cache hits to the total memory accesses that...
system_peer_cluster_cache_hit_ratio	System Peer Cluster Cache Hit Ratio	This metric measures the ratio of peer cluster cache hits to the total memory accesses that...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

system_l3_cache_hit_ratio, System L3 Cache Hit Ratio, metric

This metric measures the ratio of system L3 cache hits to the total memory accesses that missed in the private L2 cache of the core. This metric indicates that the workload is memory bound obtaining data from L3 cache.

Units

This unit is expressed as per l2 cache refill.

Formula

L3D_CACHE_HIT_RD / (L2D_CACHE_REFILL_RD + L2I_CACHE_REFILL)

Related telemetry artifacts

Events

L2D_CACHE_REFILL_RD
L2I_CACHE_REFILL

L3D_CACHE_HIT_RD

Metric group

System_Memory_Effectiveness

Methodology

Stage 2

system_peer_cluster_cache_hit_ratio, System Peer Cluster Cache Hit Ratio, metric

This metric measures the ratio of peer cluster cache hits to the total memory accesses that missed in the private L2 cache of the core. This metric indicates that the workload is memory bound obtaining data from peer cores in the compute cluster.

Units

This unit is expressed as per l2 cache refill.

Formula

(DSNP_HIT + ISNP_HIT_RD) / (L2D_CACHE_REFILL + L2I_CACHE_REFILL)

Related telemetry artifacts

Events

DSNP_HIT
ISNP_HIT_RD
L2D_CACHE_REFILL
L2I_CACHE_REFILL

Metric group

System_Memory_Effectiveness

Methodology

Stage 2

5.27 Atomics_Effectiveness metrics for C1-Nano

Atomics Effectiveness. This metric group contains metrics to evaluate the effectiveness of atomics execution on this processor.

Summary of metrics in Atomics_Effectiveness:

- Total metrics: 6

Table 5-27: Atomics_Effectiveness metrics summary

Metric	Name	Description
cas_far_ratio	Compare and Swap Far Ratio	This metric measures the ratio of compare and swap instructions that did not execute locally to...
cas_near_pass_ratio	Compare and Swap Near Pass Ratio	This metric measures the ratio of passed compare and swap instructions speculatively executed...
cas_near_ratio	Compare and Swap Near Ratio	This metric measures the ratio of compare and swap instructions speculatively executed locally to...

Metric	Name	Description
lse_atomics_ratio	LSE Atomics Ratio	This metric measures the ratio of LSE atomics instructions speculatively executed locally to the...
lse_load_ratio	LSE Load Ratio	This metric measures the ratio of LSE load instructions speculatively executed to the total LSE...
lse_store_ratio	LSE Store Ratio	This metric measures the ratio of LSE store instructions speculatively executed to the total LSE...

For a complete list of the metrics in C1-Nano, see Metrics cheat sheet for C1-Nano and Metrics lookup table for C1-Nano.

cas_far_ratio, Compare and Swap Far Ratio, metric

This metric measures the ratio of compare and swap instructions that did not execute locally to the PE to the total compare and swap instructions.

Units

This unit is expressed as per compare and swap instruction.

Formula

1 - CAS_NEAR_SPEC / CAS_SPEC

Related telemetry artifacts

Events

CAS_NEAR_SPEC
CAS_SPEC

Metric group

Atomics_Effectiveness

Methodology

Stage 2

cas_near_pass_ratio, Compare and Swap Near Pass Ratio, metric

This metric measures the ratio of passed compare and swap instructions speculatively executed locally to the PE that updated the location accessed to the total near compare and swap instructions.

Units

This unit is expressed as per near compare and swap instruction.

Formula

CAS_NEAR_PASS / CAS_NEAR_SPEC

Related telemetry artifacts

Events

CAS_NEAR_PASS
CAS_NEAR_SPEC

Metric group

Atomics_Effectiveness

Methodology

Stage 2

cas_near_ratio, Compare and Swap Near Ratio, metric

This metric measures the ratio of compare and swap instructions speculatively executed locally to the PE to the total compare and swap instructions.

Units

This unit is expressed as per compare and swap instruction.

Formula

CAS_NEAR_SPEC / CAS_SPEC

Related telemetry artifacts

Events

CAS_NEAR_SPEC
CAS_SPEC

Metric group

Atomics_Effectiveness

Methodology

Stage 2

lse_atomics_ratio, LSE Atomics Ratio, metric

This metric measures the ratio of LSE atomics instructions speculatively executed locally to the PE to the total load and store instructions.

Units

This unit is expressed as per load/store instruction.

Formula

LSE_LDST_SPEC / LDST_SPEC

Related telemetry artifacts

Events

LDST_SPEC
LSE_LDST_SPEC

Metric group

Atomics_Effectiveness

Methodology

Stage 2

lse_load_ratio, LSE Load Ratio, metric

This metric measures the ratio of LSE load instructions speculatively executed to the total LSE instructions.

Units

This unit is expressed as per lse instruction.

Formula

LSE_LD_SPEC / LSE_LDST_SPEC

Related telemetry artifacts

Events

LSE_LDST_SPEC

LSE_LD_SPEC

Metric group

Atomics_Effectiveness

Methodology

Stage 2

lse_store_ratio, LSE Store Ratio, metric

This metric measures the ratio of LSE store instructions speculatively executed to the total LSE instructions.

Units

This unit is expressed as per lse instruction.

Formula

LSE_ST_SPEC / LSE_LDST_SPEC

Related telemetry artifacts

Events

LSE_LDST_SPEC

LSE_ST_SPEC

Metric group

Atomics_Effectiveness

Methodology

Stage 2

6. PMU events by functional group in C1-Nano

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.

C1-Nano provides the following types of PMU events:

- Total implemented Common events: 308
- Total Implemented Product ImpDef events: 89
- PMU Only events : 89
- ETE Only events : 2
- Total implemented events without groups: 18

PMU events for C1-Nano are grouped into the following functional groups:

- Bus, BUS (8 events)
- Chain, CHAIN (1 events)
- Exception, EXCEPTION (4 events)
- L1D_Cache, L1D CACHE (28 events)
- L1I_Cache, L1I CACHE (8 events)
- L2_Cache, L2 CACHE (58 events)
- L3_Cache, L3 CACHE (20 events)
- LL_Cache, LL CACHE (7 events)
- Memory, MEMORY (17 events)
- Retired, RETIRED (44 events)
- Spec_Operation, SPEC OPERATION (51 events)
- Stall, STALL (48 events)
- General, GENERAL (9 events)
- TLB, TLB (29 events)
- SVE, SVE (21 events)
- Non_PMU, NON_PMU (4 events)
- TRCEXT, TRCEXT (8 events)
- Coherency, COHERENCY” (15 events)
- Events without a functional group (18 events)

6.1 Bus (BUS) events for C1-Nano

Bus transaction related events.

Summary of events in Bus:

- Total implemented Common events: 8
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-1: Bus events summary

Code	Mnemonic	Name	Description
0x0019	BUS_ACCESS	Bus access	Counts memory transactions issued by the CPU to the external bus. Each beat of data is counted...
0x001D	BUS_CYCLES	Bus cycle	Counts bus cycles in the CPU. Bus cycles represent a clock cycle in which a transaction could be...
0x0060	BUS_ACCESS_RD	Bus access, read	Counts memory read transactions seen on the external bus. Each beat of data is counted individually.
0x0061	BUS_ACCESS_WR	Bus access, write	Counts memory write transactions seen on the external bus. Each beat of data is counted...
0x8125	BUS_REQ_RD_PERCYC	Bus read transactions in progress	The counter counts by the number of bus read transactions counted by BUS_REQ_RD in progress on...
0x818D	BUS_REQ_RD	Bus request, read	Counts memory read requests issued by the complex to the downstream memory system.
0x818E	BUS_REQ_WR	Bus request, write	Counts memory write transaction requests issued by the CPU to the external bus.
0x818F	BUS_REQ	Bus request	Counts memory transaction requests issued by the CPU to the external bus.

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x0019 BUS_ACCESS, Bus access, event

Counts memory transactions issued by the CPU to the external bus. Each beat of data is counted individually. This event is the sum of BUS_ACCESS_RD and BUS_ACCESS_WR.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry 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 this Telemetry 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 this Telemetry 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 this Telemetry Specification.

Functional groups

Bus

0x8125 BUS_REQ_RD_PERCYC, Bus read transactions in progress, event

The counter counts by the number of bus read transactions counted by BUS_REQ_RD in progress on each cycle.

Related telemetry artifacts

Metrics

- bus_read_requests_average_latency in Average_Latency
- bus_read_requests_average_latency in Bus_Effectiveness

Metric groups

Average_Latency

Bus_Effectiveness

Functional groups

Bus

0x818D BUS_REQ_RD, Bus request, read, event

Counts memory read requests issued by the complex to the downstream memory system.

Related telemetry artifacts

Metrics

- bus_read_requests_average_latency in Average_Latency
- bus_read_requests_average_latency in Bus_Effectiveness

Metric groups

Average_Latency

Bus_Effectiveness

Functional groups

Bus

`0x818E` BUS_REQ_WR, Bus request, write, event

Counts memory write transaction requests issued by the CPU to the external bus.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Bus

`0x818F` BUS_REQ, Bus request, event

Counts memory transaction requests issued by the CPU to the external bus.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Bus

6.2 Chain (CHAIN) events for C1-Nano

Chain related events.

Summary of events in Chain:

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

Table 6-2: Chain events summary

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

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

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 this Telemetry Specification.

Functional groups

Chain

6.3 Exception (EXCEPTION) events for C1-Nano

Exception related events.

Summary of events in Exception:

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

Table 6-3: Exception events summary

Code	Mnemonic	Name	Description
0x0009	EXC_TAKEN	Exception taken	Counts any taken architecturally visible exceptions such as IRQ, FIQ, SError, and other...
0x000A	EXC_RETURN	Instruction architecturally executed, Condition code check pass, exception return	Counts any architecturally executed exception return instructions. For example: AArch64: ERET
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.

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

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 this Telemetry Specification.

Functional groups

Exception

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

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry 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 this Telemetry 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 this Telemetry Specification.

Functional groups

Exception

6.4 L1D_Cache (L1D CACHE) events for C1-Nano

L1 data cache related events.

Summary of events in L1D_Cache:

- Total implemented Common events: 28

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

Table 6-4: L1D_Cache events summary

Code	Mnemonic	Name	Description
0x0003	L1D_CACHE_REFILL	Level 1 data cache refill	Counts L1 data cache refills caused by speculatively executed load or store operations that...
0x0004	L1D_CACHE	Level 1 data cache access	Counts L1 data cache accesses from any load/store operations. Atomic operations that resolve in...
0x0015	L1D_CACHE_WB	Level 1 data cache write-back	Counts write-backs of data from the L1 data cache to the L2 cache. This occurs when either a...
0x0039	L1D_CACHE_LMISS_RD	Level 1 data cache long-latency read miss	Counts cache line refills into the L1 data cache from any memory read operations, that incurred...
0x0040	L1D_CACHE_RD	Level 1 data cache access, read	Counts L1 data cache accesses from any load operation. Atomic load operations that resolve in the...
0x0041	L1D_CACHE_WR	Level 1 data cache access, write	Counts L1 data cache accesses generated by store operations. This event also counts accesses...
0x0042	L1D_CACHE_REFILL_RD	Level 1 data cache refill, read	Counts L1 data cache refills caused by speculatively executed load instructions where the memory...
0x0043	L1D_CACHE_REFILL_WR	Level 1 data cache refill, write	Counts L1 data cache refills caused by speculatively executed store instructions where the memory...
0x0044	L1D_CACHE_REFILL_INNER	Level 1 data cache refill, inner	Counts L1 data cache refills where the cache line data came from caches inside the immediate...
0x0045	L1D_CACHE_REFILL_OUTER	Level 1 data cache refill, outer	Counts L1 data cache refills for which the cache line data came from outside the immediate...
0x0046	L1D_CACHE_WB_VICTIM	Level 1 data cache write-back, victim	Counts cache line evictions from the L1 data cache caused by a new cache line allocation. This...
0x8140	L1D_CACHE_RW	Level 1 data cache demand access	Counts L1 data demand cache accesses from any load or store operation. Near atomic operations...
0x8142	L1D_CACHE_PRFM	Level 1 data cache software preload	Counts L1 data cache accesses from software preload or prefetch instructions.
0x8146	L1D_CACHE_REFILL_PRFM	Level 1 data cache refill, software preload	Counts L1 data cache refills where the cache line access was generated by software preload or...
0x8154	L1D_CACHE_HWPRF	Level 1 data cache hardware prefetch	Counts L1 data cache accesses from any load/store operations generated by the hardware prefetcher.

Code	Mnemonic	Name	Description
0x81BC	L1D_CACHE_REFILL_HWPRF	Level 1 data cache refill, hardware prefetch	Counts L1 data cache refills where the cache line is requested by a hardware prefetcher.
0x81C4	L1D_CACHE_HIT_RD	Level 1 data cache demand hit, read	Counts cache line hits in the L1 data cache for memory read operations.
0x81C8	L1D_CACHE_HIT_WR	Level 1 data cache demand access hit, write	Counts cache line hits in the L1 data cache for memory write operations.
0x81CC	L1D_CACHE_HIT_RW	Level 1 data cache demand access hit	Counts cache line hits in the L1 data cache for memory read and write operations.
0x8204	L1D_CACHE_HIT	Level 1 data cache hit	Counts cache line hits in the L1 data cache.
0x8244	L1D_LFB_HIT_RD	Level 1 data cache demand line-fill buffer hit, read	Counts load data accesses that access the L1 data cache and hit in a line that is in the process...
0x8248	L1D_LFB_HIT_WR	Level 1 data cache demand access line-fill buffer hit, write	Counts store data accesses that access the L1 data cache and hit in a line that is in the process...
0x824C	L1D_LFB_HIT_RW	Level 1 data cache demand access line-fill buffer hit	Counts load or store data accesses that access the L1 data cache and hit in a line that is in the...
0x8264	L1D_LFB_HIT_RD_FHWPRF	Level 1 data cache demand line-fill buffer first hit, read, recently fetched by hardware prefetcher	Counts the first load data access that accesses the L1 data cache and hits in a line that is in...
0x8268	L1D_LFB_HIT_WR_FHWPRF	Level 1 data cache demand access line-fill buffer first hit, write, recently fetched by hardware prefetcher	Counts the first store data access that accesses the L1 data cache and hits in a line that is in...
0x826C	L1D_LFB_HIT_RW_FHWPRF	Level 1 data cache demand access line-fill buffer first hit, recently fetched by hardware prefetcher	Counts first load or store data accesses that access the L1 data cache and hit in a line that is...
0x8284	L1D_CACHE_PRF	Level 1 data cache, preload or prefetch hit	Counts L1 data cache accesses from software preload or prefetch instructions or hardware prefetcher.
0x828C	L1D_CACHE_REFILL_PRF	Level 1 data cache refill, preload or prefetch hit	Counts L1 data cache refills where the cache line is requested by software preload or prefetch...

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x0003 L1D_CACHE_REFILL, Level 1 data cache refill, event

Counts L1 data cache refills caused by speculatively executed load or store operations that missed in the L1 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 this Telemetry Specification.

Functional groups

L1D_Cache

0x0004 L1D_CACHE, Level 1 data cache access, event

Counts L1 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 L1 data or unified memory structures, for example refill buffers, write buffers, and write-back buffers, are also counted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

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

Counts write-backs of 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 this Telemetry Specification.

Functional groups

L1D_Cache

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

Counts cache line refills into the L1 data cache from any memory read operations, that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

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

Counts L1 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 this Telemetry Specification.

Functional groups

L1D_Cache

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

Counts L1 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 this Telemetry Specification.

Functional groups

L1D_Cache

0x0042 L1D_CACHE_REFILL_RD, Level 1 data cache refill, read, event

Counts L1 data cache refills caused by speculatively executed load instructions where the memory read operation misses in the L1 data cache. This event only counts one event per cache line.

Related telemetry artifacts

Metrics

- l1d_cache_mpki in L1D_Cache_Effectiveness
- l1d_cache_mpki in MPKI
- l1d_cache_miss_ratio in L1D_Cache_Effectiveness
- l1d_cache_miss_ratio in Miss_Ratio

Metric groups

L1D_Cache_Effectiveness
MPKI
Miss_Ratio

Functional groups

L1D_Cache

0x0043 L1D_CACHE_REFILL_WR, Level 1 data cache refill, write, event

Counts L1 data cache refills caused by speculatively executed store instructions where the memory write operation misses in the L1 data cache. This event only counts one event per cache line.

Related telemetry artifacts

Metrics

- l1d_cache_mpki in L1D_Cache_Effectiveness
- l1d_cache_mpki in MPKI
- l1d_cache_miss_ratio in L1D_Cache_Effectiveness
- l1d_cache_miss_ratio in Miss_Ratio

Metric groups

L1D_Cache_Effectiveness
MPKI
Miss_Ratio

Functional groups

L1D_Cache

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

Counts L1 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 this Telemetry Specification.

Functional groups

L1D_Cache

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

Counts L1 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 this Telemetry Specification.

Functional groups

L1D_Cache

`0x0046` L1D_CACHE_WB_VICTIM, Level 1 data cache write-back, victim, event

Counts cache line evictions from the L1 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 this Telemetry Specification.

Functional groups

L1D_Cache

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

Counts L1 data demand cache accesses from any load or store operation. Near atomic operations that resolve in the CPUs caches counts as both a write access and read access.

Related telemetry artifacts

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

`0x8142` L1D_CACHE_PRFM, Level 1 data cache software preload, event

Counts L1 data cache accesses from software preload or prefetch instructions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

`0x8146` L1D_CACHE_REFILL_PRFM, Level 1 data cache refill, software preload, event

Counts L1 data cache refills where the cache line access was generated by software preload or prefetch instructions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

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

Counts L1 data cache accesses from any load/store operations generated by the hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

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

Counts L1 data cache refills where the cache line is requested by a hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x81C4 L1D_CACHE_HIT_RD, Level 1 data cache demand hit, read, event

Counts cache line hits in the L1 data cache for memory read operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x81C8 L1D_CACHE_HIT_WR, Level 1 data cache demand access hit, write, event

Counts cache line hits in the L1 data cache for memory write operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x81CC L1D_CACHE_HIT_RW, Level 1 data cache demand access hit, event

Counts cache line hits in the L1 data cache for memory read and write operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x8204 L1D_CACHE_HIT, Level 1 data cache hit, event

Counts cache line hits in the L1 data cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x8244 L1D_LFB_HIT_RD, Level 1 data cache demand line-fill buffer hit, read, event

Counts load data accesses that access the L1 data cache and hit in a line that is in the process of being loaded into the L1 data cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x8248 L1D_LFB_HIT_WR, Level 1 data cache demand access line-fill buffer hit, write, event

Counts store data accesses that access the L1 data cache and hit in a line that is in the process of being loaded into the L1 data cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x824C L1D_LFB_HIT_RW, Level 1 data cache demand access line-fill buffer hit, event

Counts load or store data accesses that access the L1 data cache and hit in a line that is in the process of being loaded into the L1 data cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x8264 L1D_LFB_HIT_RD_FHWPRF, Level 1 data cache demand line-fill buffer first hit, read, recently fetched by hardware prefetcher, event

Counts the first load data access that accesses the L1 data cache and hits in a line that is in the process of being loaded into the L1 data cache by a hardware prefetch.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x8268 L1D_LFB_HIT_WR_FHWPRF, Level 1 data cache demand access line-fill buffer first hit, write, recently fetched by hardware prefetcher, event

Counts the first store data access that accesses the L1 data cache and hits in a line that is in the process of being loaded into the L1 data cache by a hardware prefetch.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x826C L1D_LFB_HIT_RW_FHWPRF, Level 1 data cache demand access line-fill buffer first hit, recently fetched by hardware prefetcher, event

Counts first load or store data accesses that access the L1 data cache and hit in a line that is in the process of being loaded into the L1 data cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x8284 L1D_CACHE_PRF, Level 1 data cache, preload or prefetch hit, event

Counts L1 data cache accesses from software preload or prefetch instructions or hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

0x828C L1D_CACHE_REFILL_PRF, Level 1 data cache refill, preload or prefetch hit, event

Counts L1 data cache refills where the cache line is requested by software preload or prefetch instructions or hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1D_Cache

6.5 L1I_Cache (L1I CACHE) events for C1-Nano

L1 instruction cache related events.

Summary of events in L1I_Cache:

- Total implemented Common events: 8
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-5: L1I_Cache events summary

Code	Mnemonic	Name	Description
0x0001	L1I_CACHE_REFILL	Level 1 instruction cache refill	Counts cache line refills in the L1 instruction cache caused by a missed instruction fetch....
0x0014	L1I_CACHE	Level 1 instruction cache access	Counts instruction fetches which access the L1 instruction cache. Instruction cache accesses...
0x4006	L1I_CACHE_LMISS	Level 1 instruction cache long-latency miss	Counts cache line refills into the L1 instruction cache, that incurred additional latency.
0x8141	L1I_CACHE_RD	Level 1 instruction cache demand fetch	Counts demand instruction fetches which access the L1 instruction cache.
0x8145	L1I_CACHE_HWPRF	Level 1 instruction cache hardware prefetch	Counts instruction fetches which access the L1 instruction cache generated by the hardware...
0x81C0	L1I_CACHE_HIT_RD	Level 1 instruction cache demand fetch hit	Counts demand instruction fetches that access the L1 instruction cache and hit in the L1...
0x8200	L1I_CACHE_HIT	Level 1 instruction cache hit	Counts instruction fetches that access the L1 instruction cache and hit in the L1 instruction...
0x8240	L1I_LFB_HIT_RD	Level 1 instruction cache demand fetch line-fill buffer hit	Counts demand instruction fetches that access the L1 instruction cache and hit in a line that is...

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x0001 L1I_CACHE_REFILL, Level 1 instruction cache refill, event

Counts cache line refills in the L1 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 L1 instruction cache. Instruction cache accesses caused by cache maintenance operations are not counted.

Related telemetry artifacts

Metrics

- l1i_cache_miss_ratio in L1I_Cache_Effectiveness
- l1i_cache_miss_ratio in Miss_Ratio

Metric groups

L1I_Cache_Effectiveness
Miss_Ratio

Functional groups

L1I_Cache

`0x4006` L1I_CACHE_LMISS, Level 1 instruction cache long-latency miss, event

Counts cache line refills into the L1 instruction cache, that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1I_Cache

`0x8141` L1I_CACHE_RD, Level 1 instruction cache demand fetch, event

Counts demand instruction fetches which access the L1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1I_Cache

`0x8145` L1I_CACHE_HWPRF, Level 1 instruction cache hardware prefetch, event

Counts instruction fetches which access the L1 instruction cache generated by the hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1I_Cache

`0x81C0` L1I_CACHE_HIT_RD, Level 1 instruction cache demand fetch hit, event

Counts demand instruction fetches that access the L1 instruction cache and hit in the L1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1I_Cache

0x8200 L1I_CACHE_HIT, Level 1 instruction cache hit, event

Counts instruction fetches that access the L1 instruction cache and hit in the L1 instruction cache. Instruction cache accesses caused by cache maintenance operations are not counted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1I_Cache

0x8240 L1I_LFB_HIT_RD, Level 1 instruction cache demand fetch line-fill buffer hit, event

Counts demand instruction fetches that access the L1 instruction cache and hit in a line that is in the process of being loaded into the L1 instruction cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L1I_Cache

6.6 L2_Cache (L2 CACHE) events for C1-Nano

L2 unified cache related events.

Summary of events in L2_Cache:

- Total implemented Common events: 36
- Total Implemented Product ImpDef events: 22
- PMU Only events : 22
- ETE Only events : 0

Table 6-6: L2_Cache events summary

Code	Mnemonic	Name	Description
0x0016	L2D_CACHE	Level 2 data cache access	Counts L2 cache accesses. L2 cache is a unified cache for data and instruction accesses. Accesses...
0x0017	L2D_CACHE_REFILL	Level 2 data cache refill	Counts L2 cache refills. L2 cache is a unified cache for data and instruction accesses. Accesses...

Code	Mnemonic	Name	Description
0x0018	L2D_CACHE_WB	Level 2 data cache write-back	Counts write-backs of data from the L2 cache to outside the CPU as a result of either: • a...
0x0020	L2D_CACHE_ALLOCATE	Level 2 data cache allocation without refill	Counts memory write operations that write a full cache line into the L2 cache without fetching...
0x0027	L2I_CACHE	Level 2 instruction cache access	Counts accesses to the L2 cache due to instruction accesses. L2 cache is a unified cache for data...
0x0028	L2I_CACHE_REFILL	Level 2 instruction cache refill	Counts L2 cache refills due to instruction accesses. L2 cache is a unified cache for data and...
0x0050	L2D_CACHE_RD	Level 2 data cache access, read	Counts L2 cache accesses due to memory read operations. level 2 cache is a unified cache for data...
0x0051	L2D_CACHE_WR	Level 2 data cache access, write	Counts L2 cache accesses due to memory write operations. level 2 cache is a unified cache for...
0x0052	L2D_CACHE_REFILL_RD	Level 2 data cache refill, read	Counts L2 cache refills due to demand memory read operations. level 2 cache is a unified cache...
0x0053	L2D_CACHE_REFILL_WR	Level 2 data cache refill, write	Counts L2 cache refills due to memory write operations. level 2 cache is a unified cache for data...
0x0056	L2D_CACHE_WB_VICTIM	Level 2 data cache write-back, victim	Counts evictions from the L2 cache because of a line being allocated into the L2 cache.
0x00d9	IMP_L2D_CACHE_REFILL_HWPRF_CORRELATING	L2 cache refill, hardware correlating prefetcher access	Counts L2 cache refills for accesses generated by the hardware correlating prefetcher, targeting...
0x00da	IMP_L2D_CACHE_REFILL_HWPRF_SPATIAL	L2 cache refill, hardware spatial prefetcher access	Counts L2 cache refills for accesses generated by the hardware spatial prefetcher, targeting the...
0x00db	IMP_L2D_CACHE_REFILL_HWPRF_OFFSET	L2 cache refill, hardware offset prefetcher access	Counts L2 cache refills for accesses generated by the hardware offset prefetcher, targeting the...
0x0100	IMP_L2D_CACHE_HIT_HWPRF	L2 cache hit, hardware prefetch access	Counts L2 cache hits for accesses generated by hardware prefetchers targeting the L2 cache.
0x0101	IMP_L2D_CACHE_HIT_L1DHWPRF	L2 cache hit on access, L1D hardware prefetcher	Counts L2 cache hits from L1D hardware prefetches.
0x0102	IMP_L2D_CACHE_HIT_L1DHWPRF_FHWPRF	L2 cache first hit on access, L1D hardware prefetcher	Counts L2 cache hits due to L1 hardware prefetcher accesses which are the first hit on a cache...
0x0103	IMP_L2D_CACHE_HIT_PRF	L2 cache hit, software or hardware prefetch access	Counts L2 cache hits counted by either IMP_L2D_CACHE_HIT_HWPRF or L2D_CACHE_HIT_PRFM.

Code	Mnemonic	Name	Description
0x0104	IMP_L2D_CACHE_HIT_RD_FHWPRF_CORRELATING	L2 cache first hit on prefetched line, hardware correlating prefetcher	Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a...
0x0105	IMP_L2D_CACHE_HIT_RD_FHWPRF_OFFSET	L2 cache first hit on prefetched line, hardware offset prefetcher	Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a...
0x0106	IMP_L2D_CACHE_HIT_RD_FHWPRF_SPATIAL	L2 cache first hit on prefetched line, hardware spatial prefetcher	Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a...
0x0107	IMP_L2D_CACHE_HIT_RD_FHWPRF_STRIDE	L2 cache first hit on prefetched line, hardware stride prefetcher	Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a...
0x0108	IMP_L2D_CACHE_HIT_RD_FHWPRF_TLB	L2 cache first hit on prefetched line, hardware TLB descriptor prefetcher	Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a...
0x0109	IMP_L2D_CACHE_HWPRF_CORRELATING	L2 cache access, hardware correlating prefetcher	Counts L2 cache accesses generated by the hardware correlating prefetcher, targeting the L2 cache.
0x010a	IMP_L2D_CACHE_HWPRF_OFFSET	L2 cache access, hardware offset prefetcher	Counts L2 cache accesses generated by the hardware offset prefetcher, targeting the L2 cache.
0x010b	IMP_L2D_CACHE_HWPRF_SPATIAL	L2 cache access, hardware spatial prefetcher	Counts L2 cache accesses generated by the hardware spatial prefetcher, targeting the L2 cache.
0x010c	IMP_L2D_CACHE_HWPRF_STRIDE	L2 cache access, hardware stride prefetcher	Counts L2 cache accesses generated by the hardware stride prefetcher, targeting the L2 cache.
0x010d	IMP_L2D_CACHE_HWPRF_TLB	L2 cache access, hardware TLB descriptor prefetcher	Counts L2 cache accesses generated by the hardware TLB descriptor prefetcher, targeting the L2...
0x010e	IMP_L2D_CACHE_L1DHWPRF	L2 cache access, L1D hardware prefetcher	Counts L2 cache accesses from L1D hardware prefetches.
0x010f	IMP_L2D_CACHE_REFILL_HWPRF_STRIDE	L2 cache refill, hardware stride prefetcher access	Counts L2 cache refills for accesses generated by the hardware stride prefetcher, targeting the...
0x0110	IMP_L2D_CACHE_REFILL_HWPRF_TLB	L2 cache refill, hardware TLB descriptor prefetcher access	Counts L2 cache refills for accesses generated by the hardware TLB descriptor prefetcher,...
0x0111	IMP_L2D_CACHE_REFILL_L1DHWPRF	L2 cache refill, L1D hardware prefetcher access	Counts L2 cache refills for accesses from L1D hardware prefetches.
0x0112	IMP_L2D_LFB_HIT_L1DHWPRF_FHWPRF	L2 cache line-fill buffer hit on access, L1D hardware prefetcher	Counts L2 line-fill buffer first hits by L1 hardware prefetcher accesses, where the cache line...

Code	Mnemonic	Name	Description
0x4009	L2D_CACHE_LMISS_RD	Level 2 data cache long-latency read miss	Counts cache line refills into the L2 unified cache from any memory read operations that incurred...
0x8148	L2D_CACHE_RW	Level 2 data cache demand access	Counts L2 cache demand accesses from any load/store operations or instruction fetches. L2 cache...
0x8149	L2I_CACHE_RD	Level 2 instruction cache demand fetch	Counts L2 cache accesses that are due to a demand instruction cache access.
0x814A	L2D_CACHE_PRFM	Level 2 data cache software preload	Counts L2 cache accesses generated by software preload or prefetch instructions targeting the L2...
0x814E	L2D_CACHE_REFILL_PRFM	Level 2 data cache refill, software preload	Counts L2 cache refills for accesses generated by software preload or prefetch instructions...
0x8155	L2D_CACHE_HWPRF	Level 2 data cache hardware prefetch	Counts L2 cache accesses generated by hardware prefetchers targeting the L2 cache.
0x81BD	L2D_CACHE_REFILL_HWPRF	Level 2 data cache refill, hardware prefetch	Counts L2 cache refills for accesses generated by hardware prefetchers targeting the L2 cache.
0x81C1	L2I_CACHE_HIT_RD	Level 2 instruction cache demand fetch hit	Counts L2I cache accesses that hit in the L2 cache.
0x81C5	L2D_CACHE_HIT_RD	Level 2 data cache demand hit, read	Counts L2 cache hits due to demand memory read operations.
0x81C9	L2D_CACHE_HIT_WR	Level 2 data cache demand access hit, write	Counts L2 cache hits due to memory write operations.
0x81CD	L2D_CACHE_HIT_RW	Level 2 data cache demand access hit	Counts L2 cache demand hits from any load/store operations or instruction fetches.
0x81E5	L2D_CACHE_HIT_RD_FHWPRF	Level 2 data cache demand first hit, read, fetched by hardware prefetcher	Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a...
0x81E9	L2D_CACHE_HIT_WR_FHWPRF	Level 2 data cache demand access first hit, write, fetched by hardware prefetcher	Counts L2 cache hits due to demand memory write operations, which are the first demand hit on a...
0x81ED	L2D_CACHE_HIT_RW_FHWPRF	Level 2 data cache demand access first hit, fetched by hardware prefetcher	Counts L2 cache hits due to demand memory operations, which are the first demand hit on a cache...
0x8201	L2I_CACHE_HIT	Level 2 instruction cache hit	-
0x8205	L2D_CACHE_HIT	Level 2 data cache hit	Counts L2 cache hits. For the L2D_CACHE_HIT events, an L2 cache hit is only counted if the...
0x820D	L2D_CACHE_HIT_PRFM	Level 2 data cache software preload hit	Counts L2 cache hits for accesses generated by prefetch instructions targeting the L2 cache.

Code	Mnemonic	Name	Description
0x8245	L2D_LFB_HIT_RD	Level 2 data cache demand line-fill buffer hit, read	The counter counts each demand Memory-read operation that hits a recently fetched line that is in...
0x8249	L2D_LFB_HIT_WR	Level 2 data cache demand access line-fill buffer hit, write	The counter counts each demand Memory-write operation that hits a recently fetched line that is...
0x824D	L2D_LFB_HIT_RW	Level 2 data cache demand access line-fill buffer hit	The counter counts each demand access that hits a recently fetched line that is in the process of...
0x8265	L2D_LFB_HIT_RD_FHWPRF	Level 2 data cache demand line-fill buffer first hit, read, recently fetched by hardware prefetcher	The counter counts each demand Memory-read operation line-fill buffer first hit counted by...
0x8269	L2D_LFB_HIT_WR_FHWPRF	Level 2 data cache demand access line-fill buffer first hit, write, recently fetched by hardware prefetcher	The counter counts each demand Memory-write operation line-fill buffer first hit counted by...
0x826D	L2D_LFB_HIT_RW_FHWPRF	Level 2 data cache demand access line-fill buffer first hit, recently fetched by hardware prefetcher	The counter counts each demand access line-fill buffer first hit counted by L2D_LFB_HIT_RW where...
0x8285	L2D_CACHE_PRF	Level 2 data cache, preload or prefetch hit	Counts L2 cache accesses counted by either L2D_CACHE_HWPRF or L2D_CACHE_PRFM.
0x828D	L2D_CACHE_REFILL_PRF	Level 2 data cache refill, preload or prefetch hit	Counts L2 cache refills counted by either L2D_CACHE_REFILL_HWPRF or L2D_CACHE_REFILL_PRFM.

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x0016 L2D_CACHE, Level 2 data cache access, event

Counts L2 cache accesses. L2 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 L1 data cache to the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x0017 L2D_CACHE_REFILL, Level 2 data cache refill, event

Counts L2 cache refills. L2 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 L1 data cache to the L2 cache.

Related telemetry artifacts

Metrics

- system_peer_cluster_cache_hit_ratio

Metric groups

System_Memory_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 as a result of either:

- a line being allocated into the L2 cache which then causes a line to be evicted
- cache maintenance operations

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

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

Counts memory write operations that write a full cache line into the L2 cache without fetching data from outside the L2 cache. These are allocations of cache lines in the L2 cache that are not refills counted by L2D_CACHE_REFILL.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x0027 L2I_CACHE, Level 2 instruction cache access, event

Counts accesses to the L2 cache due to instruction accesses. L2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the first level instruction cache.

Related telemetry artifacts

Metrics

- l2i_cache_miss_ratio in L2I_Cache_Effectiveness
- l2i_cache_miss_ratio in Miss_Ratio

Metric groups

L2I_Cache_Effectiveness
Miss_Ratio

Functional groups

L2_Cache

0x0028 L2I_CACHE_REFILL, Level 2 instruction cache refill, event

Counts L2 cache refills due to instruction accesses. L2 cache is a unified cache for data and instruction accesses. Accesses are for misses in the L1 instruction cache.

Related telemetry artifacts

Metrics

- l2i_cache_mpki in L2I_Cache_Effectiveness
- l2i_cache_mpki in MPKI
- l2i_cache_miss_ratio in L2I_Cache_Effectiveness
- l2i_cache_miss_ratio in Miss_Ratio
- system_l3_cache_hit_ratio
- system_peer_cluster_cache_hit_ratio

Metric groups

L2I_Cache_Effectiveness
MPKI
Miss_Ratio
System_Memory_Effectiveness

Functional groups

L2_Cache

0x0050 L2D_CACHE_RD, Level 2 data cache access, read, event

Counts L2 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 L1 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 this Telemetry Specification.

Functional groups

L2_Cache

0x0051 L2D_CACHE_WR, Level 2 data cache access, write, event

Counts L2 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 L1 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 this Telemetry Specification.

Functional groups

L2_Cache

0x0052 L2D_CACHE_REFILL_RD, Level 2 data cache refill, read, event

Counts L2 cache refills due to demand memory read operations. level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the L1 caches or translation resolutions due to accesses.

Related telemetry artifacts

Metrics

- l2d_cache_mpki in L2D_Cache_Effectiveness
- l2d_cache_mpki in MPKI
- l2d_cache_miss_ratio in L2D_Cache_Effectiveness
- l2d_cache_miss_ratio in Miss_Ratio
- l2_prefetcher_coverage_l1hwprf_exclusive
- system_l3_cache_hit_ratio

Metric groups

L2D_Cache_Effectiveness
MPKI
Miss_Ratio
Prefetcher_Effectiveness
System_Memory_Effectiveness

Functional groups

L2_Cache

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

Counts L2 cache refills due to memory write operations. level 2 cache is a unified cache for data and instruction accesses, accesses are for misses in the L1 caches or translation resolutions due to accesses.

Related telemetry artifacts

Metrics

- l2d_cache_mpki in L2D_Cache_Effectiveness
- l2d_cache_mpki in MPKI
- l2d_cache_miss_ratio in L2D_Cache_Effectiveness
- l2d_cache_miss_ratio in Miss_Ratio
- l2_prefetcher_coverage_l1hwprf_exclusive

Metric groups

L2D_Cache_Effectiveness
MPKI

Miss_Ratio
Prefetcher_Effectiveness

Functional groups

L2_Cache

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

Counts evictions from the L2 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 this Telemetry Specification.

Functional groups

L2_Cache

0x00d9 IMP_L2D_CACHE_REFILL_HWPRF_CORRELATING, L2 cache refill, hardware correlating prefetcher access, event

Counts L2 cache refills for accesses generated by the hardware correlating prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x00da IMP_L2D_CACHE_REFILL_HWPRF_SPATIAL, L2 cache refill, hardware spatial prefetcher access, event

Counts L2 cache refills for accesses generated by the hardware spatial prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x00db IMP_L2D_CACHE_REFILL_HWPRF_OFFSET, L2 cache refill, hardware offset prefetcher access, event

Counts L2 cache refills for accesses generated by the hardware offset prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x0100 IMP_L2D_CACHE_HIT_HWPRF, L2 cache hit, hardware prefetch access, event

Counts L2 cache hits for accesses generated by hardware prefetchers targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x0101 IMP_L2D_CACHE_HIT_L1DHWPRF, L2 cache hit on access, L1D hardware prefetcher, event

Counts L2 cache hits from L1D hardware prefetches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x0102 IMP_L2D_CACHE_HIT_L1DHWPRF_FHWPRF, L2 cache first hit on access, L1D hardware prefetcher, event

Counts L2 cache hits due to L1 hardware prefetcher accesses which are the first hit on a cache line prefetched by an L2 hardware prefetcher into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x0103 IMP_L2D_CACHE_HIT_PRF, L2 cache hit, software or hardware prefetch access, event

Counts L2 cache hits counted by either IMP_L2D_CACHE_HIT_HWPRF or L2D_CACHE_HIT_PRFM.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x0104 IMP_L2D_CACHE_HIT_RD_FHWPRF_CORRELATING, L2 cache first hit on prefetched line, hardware correlating prefetcher, event

Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a cache line prefetched by the hardware correlating prefetcher into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x0105` IMP_L2D_CACHE_HIT_RD_FHWPRF_OFFSET, L2 cache first hit on prefetched line, hardware offset prefetcher, event

Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a cache line prefetched by the hardware offset prefetcher into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x0106` IMP_L2D_CACHE_HIT_RD_FHWPRF_SPATIAL, L2 cache first hit on prefetched line, hardware spatial prefetcher, event

Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a cache line prefetched by the hardware spatial prefetcher into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x0107` IMP_L2D_CACHE_HIT_RD_FHWPRF_STRIDE, L2 cache first hit on prefetched line, hardware stride prefetcher, event

Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a cache line prefetched by the hardware stride prefetcher into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x0108` IMP_L2D_CACHE_HIT_RD_FHWPRF_TLB, L2 cache first hit on prefetched line, hardware TLB descriptor prefetcher, event

Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a cache line prefetched by the hardware TLB descriptor prefetcher into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x0109` IMP_L2D_CACHE_HWPRF_CORRELATING, L2 cache access, hardware correlating prefetcher, event

Counts L2 cache accesses generated by the hardware correlating prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x010a` IMP_L2D_CACHE_HWPRF_OFFSET, L2 cache access, hardware offset prefetcher, event

Counts L2 cache accesses generated by the hardware offset prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x010b` IMP_L2D_CACHE_HWPRF_SPATIAL, L2 cache access, hardware spatial prefetcher, event

Counts L2 cache accesses generated by the hardware spatial prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x010c` IMP_L2D_CACHE_HWPRF_STRIDE, L2 cache access, hardware stride prefetcher, event

Counts L2 cache accesses generated by the hardware stride prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x010d` IMP_L2D_CACHE_HWPRF_TLB, L2 cache access, hardware TLB descriptor prefetcher, event

Counts L2 cache accesses generated by the hardware TLB descriptor prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x010e` IMP_L2D_CACHE_L1DHWPRF, L2 cache access, L1D hardware prefetcher, event

Counts L2 cache accesses from L1D hardware prefetches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x010f` IMP_L2D_CACHE_REFILL_HWPRF_STRIDE, L2 cache refill, hardware stride prefetcher access, event

Counts L2 cache refills for accesses generated by the hardware stride prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x0110` IMP_L2D_CACHE_REFILL_HWPRF_TLB, L2 cache refill, hardware TLB descriptor prefetcher access, event

Counts L2 cache refills for accesses generated by the hardware TLB descriptor prefetcher, targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x0111` IMP_L2D_CACHE_REFILL_L1DHWPRF, L2 cache refill, L1D hardware prefetcher access, event

Counts L2 cache refills for accesses from L1D hardware prefetches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

`0x0112` IMP_L2D_LFB_HIT_L1DHWPRF_FHWPRF, L2 cache line-fill buffer hit on access, L1D hardware prefetcher, event

Counts L2 line-fill buffer first hits by L1 hardware prefetcher accesses, where the cache line was fetched by an L2 hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

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

Counts cache line refills into the L2 unified cache from any memory read operations that incurred additional latency.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

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

Counts L2 cache demand accesses from any load/store operations or instruction fetches. L2 cache is a unified cache for data and instruction accesses, and so therefore accesses are either data accesses missed in L1 data cache or instruction fetches missed in the L1 instruction cache.

Related telemetry artifacts

Metrics

- l2d_cache_miss_ratio in L2D_Cache_Effectiveness
- l2d_cache_miss_ratio in Miss_Ratio

Metric groups

L2D_Cache_Effectiveness

Miss_Ratio

Functional groups

L2_Cache

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

Counts L2 cache accesses that are due to a demand instruction cache access.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x814A L2D_CACHE_PRFM, Level 2 data cache software preload, event

Counts L2 cache accesses generated by software preload or prefetch instructions targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x814E L2D_CACHE_REFILL_PRFM, Level 2 data cache refill, software preload, event

Counts L2 cache refills for accesses generated by software preload or prefetch instructions targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x8155 L2D_CACHE_HWPRF, Level 2 data cache hardware prefetch, event

Counts L2 cache accesses generated by hardware prefetchers targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x81BD L2D_CACHE_REFILL_HWPRF, Level 2 data cache refill, hardware prefetch, event

Counts L2 cache refills for accesses generated by hardware prefetchers targeting the L2 cache.

Related telemetry artifacts

Metrics

- l2_prefetcher_accuracy_l1hwprf_exclusive

Metric groups

Prefetcher_Effectiveness

Functional groups

L2_Cache

0x81C1 L2I_CACHE_HIT_RD, Level 2 instruction cache demand fetch hit, event

Counts L2I cache accesses that hit in the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x81C5 L2D_CACHE_HIT_RD, Level 2 data cache demand hit, read, event

Counts L2 cache hits due to demand memory read operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x81C9 L2D_CACHE_HIT_WR, Level 2 data cache demand access hit, write, event

Counts L2 cache hits due to memory write operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x81CD L2D_CACHE_HIT_RW, Level 2 data cache demand access hit, event

Counts L2 cache demand hits from any load/store operations or instruction fetches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x81E5 L2D_CACHE_HIT_RD_FHWPRF, Level 2 data cache demand first hit, read, fetched by hardware prefetcher, event

Counts L2 cache hits due to demand memory read operations, which are the first demand hit on a cache line prefetched by a hardware prefetcher into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x81E9 L2D_CACHE_HIT_WR_FHWPRF, Level 2 data cache demand access first hit, write, fetched by hardware prefetcher, event

Counts L2 cache hits due to demand memory write operations, which are the first demand hit on a cache line prefetched by a hardware prefetcher into the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x81ED L2D_CACHE_HIT_RW_FHWPRF, Level 2 data cache demand access first hit, fetched by hardware prefetcher, event

Counts L2 cache hits due to demand memory operations, which are the first demand hit on a cache line prefetched by a hardware prefetcher into the L2 cache.

Related telemetry artifacts

Metrics

- l2_prefetcher_coverage_l1hwprf_exclusive
- l2_prefetcher_accuracy_l1hwprf_exclusive
- l2_prefetcher_timeliness_l1hwprf_exclusive

Metric groups

Prefetcher_Effectiveness

Functional groups

L2_Cache

0x8201 L2I_CACHE_HIT, Level 2 instruction cache hit, event

Level 2 instruction cache hit.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x8205 L2D_CACHE_HIT, Level 2 data cache hit, event

Counts L2 cache hits. For the L2D_CACHE_HIT events, an L2 cache hit is only counted if the operation is satisfied internally by L2 (i.e. no bus request, no snoop to lower level).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x820D L2D_CACHE_HIT_PRFM, Level 2 data cache software preload hit, event

Counts L2 cache hits for accesses generated by prefetch instructions targeting the L2 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x8245 L2D_LFB_HIT_RD, Level 2 data cache demand line-fill buffer hit, read, event

The counter counts each demand Memory-read operation that hits a recently fetched line that is in the process of being loaded into the L2 data or unified cache. This means it will not generate a new refill, but has to wait for the previous refill to complete.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x8249 L2D_LFB_HIT_WR, Level 2 data cache demand access line-fill buffer hit, write, event

The counter counts each demand Memory-write operation that hits a recently fetched line that is in the process of being loaded into the L2 data or unified cache. This means it will not generate a new refill, but has to wait for the previous refill to complete.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x824D L2D_LFB_HIT_RW, Level 2 data cache demand access line-fill buffer hit, event

The counter counts each demand access that hits a recently fetched line that is in the process of being loaded into the L2 data or unified cache. This means it will not generate a new refill, but has to wait for the previous refill to complete.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x8265 L2D_LFB_HIT_RD_FHWPRF, Level 2 data cache demand line-fill buffer first hit, read, recently fetched by hardware prefetcher, event

The counter counts each demand Memory-read operation line-fill buffer first hit counted by L2D_LFB_HIT_RD where the cache line was fetched by a hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x8269 L2D_LFB_HIT_WR_FHWPRF, Level 2 data cache demand access line-fill buffer first hit, write, recently fetched by hardware prefetcher, event

The counter counts each demand Memory-write operation line-fill buffer first hit counted by L2D_LFB_HIT_WR where the cache line was fetched by a hardware prefetcher.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

0x826D L2D_LFB_HIT_RW_FHWPRF, Level 2 data cache demand access line-fill buffer first hit, recently fetched by hardware prefetcher, event

The counter counts each demand access line-fill buffer first hit counted by L2D_LFB_HIT_RW where the cache line was fetched by a hardware prefetcher.

Related telemetry artifacts

Metrics

- l2_prefetcher_coverage_l1hwprf_exclusive
- l2_prefetcher_accuracy_l1hwprf_exclusive
- l2_prefetcher_timeliness_l1hwprf_exclusive

Metric groups

Prefetcher_Effectiveness

Functional groups

L2_Cache

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

Counts L2 cache accesses counted by either L2D_CACHE_HWPRF or L2D_CACHE_PRFM.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

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

Counts L2 cache refills counted by either L2D_CACHE_REFILL_HWPRF or L2D_CACHE_REFILL_PRFM.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L2_Cache

6.7 L3_Cache (L3 CACHE) events for C1-Nano

L3 unified cache related events.

Summary of events in L3_Cache:

- Total implemented Common events: 10
- Total Implemented Product ImpDef events: 10
- PMU Only events : 10
- ETE Only events : 0

Table 6-7: L3_Cache events summary

Code	Mnemonic	Name	Description
0x002B	L3D_CACHE	Level 3 data cache access	Counts level 3 cache accesses. level 3 cache is a unified cache for data and instruction...
0x00A0	L3D_CACHE_RD	Level 3 data cache access, read	The counter counts each access counted by L3D_CACHE that is a Memory-read operation.
0x00A1	L3D_CACHE_WR	Level 3 data cache access, write	Counts level 3 cache accesses due to memory write operations.
0x00A2	L3D_CACHE_REFILL_RD	Level 3 data cache refill, read	Counts level 3 accesses that receive data from outside the L3 cache due to a Memory-read operation.
0x00de	IMP_L3D_CACHE_HWPRF_STRIDE	Level 3 cache access, hardware stride prefetcher	Counts level 3 cache accesses generated by the hardware stride prefetcher, targeting the level 3...
0x00df	IMP_L3D_CACHE_HWPRF_OFFSET	Level 3 cache access, hardware offset prefetcher	Counts level 3 cache accesses generated by the hardware offset prefetcher, targeting the level 3...
0x0120	IMP_L3D_CACHE_L1DHWPRF	Level 3 cache access, L1D hardware prefetcher	Counts level 3 cache accesses from L1D hardware prefetches.

Code	Mnemonic	Name	Description
0x0122	IMP_L3D_CACHE_REFILL_L1DHWPRF	Level 3 cache refill, L1D hardware prefetcher access	Counts level 3 cache accesses from L1D hardware prefetches that receive data from outside the L3...
0x0123	IMP_L3D_CACHE_REFILL_L2DHWPRF	Level 3 cache refill, L2D hardware prefetcher access	Counts level 3 cache accesses from L2D hardware prefetches that receive data from outside the L3...
0x0124	IMP_L3D_CACHE_REFILL_L2DHWPRF_CORRELATING	Level 3 cache refill, L2D hardware correlating prefetcher access	Counts level 3 cache accesses from L2D hardware correlating prefetches that receive data from...
0x0125	IMP_L3D_CACHE_REFILL_L2DHWPRF_OFFSET	Level 3 cache refill, L2D hardware offset prefetcher access	Counts level 3 cache accesses from L2D hardware offset prefetches that receive data from outside...
0x0126	IMP_L3D_CACHE_REFILL_L2DHWPRF_SPATIAL	Level 3 cache refill, L2D hardware spatial prefetcher access	Counts level 3 cache accesses from L2D hardware spatial prefetches that receive data from outside...
0x0127	IMP_L3D_CACHE_REFILL_L2DHWPRF_STRIDE	Level 3 cache refill, L2D hardware stride prefetcher access	Counts level 3 cache accesses from L2D hardware stride prefetches that receive data from outside...
0x0128	IMP_L3D_CACHE_REFILL_L2DHWPRF_TLB	Level 3 cache refill, L2D hardware TLB descriptor prefetcher access	Counts level 3 cache accesses from L2D hardware TLB descriptor prefetches that receive data from...
0x400B	L3D_CACHE_LMISS_RD	Level 3 data cache long-latency read miss	Counts any cache line refill into the level 3 cache from memory read operations that incurred...
0x8151	L3D_CACHE_PRFM	Level 3 data cache software preload	Counts level 3 cache accesses generated by software preload or prefetch instructions targeting...
0x8156	L3D_CACHE_HWPRF	Level 3 data cache hardware prefetch	Counts level 3 cache accesses generated by hardware prefetchers targeting the level 3 cache.
0x81C6	L3D_CACHE_HIT_RD	Level 3 data cache demand hit, read	The counter counts each demand read counted by L3D_CACHE_RD that hits in the Level 3 data or...
0x81E6	L3D_CACHE_HIT_RD_FHWPRF	Level 3 data cache demand first hit, read, fetched by hardware prefetcher	The counter counts each first hit counted by L3D_CACHE_HIT_RW_FHWPRF that is a demand Memory-read...
0x8286	L3D_CACHE_PRF	Level 3 data cache, preload or prefetch hit	Counts level 3 cache accesses counted by either L3D_CACHE_HWPRF or L3D_CACHE_PRFM.

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

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 this Telemetry Specification.

Functional groups

L3_Cache

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

The counter counts each access counted by L3D_CACHE that is a Memory-read operation.

Related telemetry artifacts

Metrics

- l3_cache_miss_ratio in L3_Cache_Effectiveness
- l3_cache_miss_ratio in Miss_Ratio

Metric groups

L3_Cache_Effectiveness
Miss_Ratio

Functional groups

L3_Cache

0x00A1 L3D_CACHE_WR, Level 3 data cache access, write, event

Counts level 3 cache accesses due to memory write operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

0x00A2 L3D_CACHE_REFILL_RD, Level 3 data cache refill, read, event

Counts level 3 accesses that receive data from outside the L3 cache due to a Memory-read operation.

Related telemetry artifacts

Metrics

- l3_cache_mpki in L3_Cache_Effectiveness
- l3_cache_mpki in MPKI
- l3_cache_miss_ratio in L3_Cache_Effectiveness
- l3_cache_miss_ratio in Miss_Ratio

Metric groups

L3_Cache_Effectiveness
MPKI
Miss_Ratio

Functional groups

L3_Cache

`0x00de` IMP_L3D_CACHE_HWPRF_STRIDE, Level 3 cache access, hardware stride prefetcher, event

Counts level 3 cache accesses generated by the hardware stride prefetcher, targeting the level 3 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

`0x00df` IMP_L3D_CACHE_HWPRF_OFFSET, Level 3 cache access, hardware offset prefetcher, event

Counts level 3 cache accesses generated by the hardware offset prefetcher, targeting the level 3 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

`0x0120` IMP_L3D_CACHE_L1DHWPRF, Level 3 cache access, L1D hardware prefetcher, event

Counts level 3 cache accesses from L1D hardware prefetches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

`0x0122` IMP_L3D_CACHE_REFILL_L1DHWPRF, Level 3 cache refill, L1D hardware prefetcher access, event

Counts level 3 cache accesses from L1D hardware prefetches 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 this Telemetry Specification.

Functional groups

L3_Cache

0x0123 IMP_L3D_CACHE_REFILL_L2DHWPRF, Level 3 cache refill, L2D hardware prefetcher access, event

Counts level 3 cache accesses from L2D hardware prefetches that receive data from outside the L3 cache. Refer to child events for a breakdown.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

0x0124 IMP_L3D_CACHE_REFILL_L2DHWPRF_CORRELATING, Level 3 cache refill, L2D hardware correlating prefetcher access, event

Counts level 3 cache accesses from L2D hardware correlating prefetches 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 this Telemetry Specification.

Functional groups

L3_Cache

0x0125 IMP_L3D_CACHE_REFILL_L2DHWPRF_OFFSET, Level 3 cache refill, L2D hardware offset prefetcher access, event

Counts level 3 cache accesses from L2D hardware offset prefetches 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 this Telemetry Specification.

Functional groups

L3_Cache

0x0126 IMP_L3D_CACHE_REFILL_L2DHWPRF_SPATIAL, Level 3 cache refill, L2D hardware spatial prefetcher access, event

Counts level 3 cache accesses from L2D hardware spatial prefetches 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 this Telemetry Specification.

Functional groups

L3_Cache

0x0127 IMP_L3D_CACHE_REFILL_L2DHWPRF_STRIDE, Level 3 cache refill, L2D hardware stride prefetcher access, event

Counts level 3 cache accesses from L2D hardware stride prefetches 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 this Telemetry Specification.

Functional groups

L3_Cache

0x0128 IMP_L3D_CACHE_REFILL_L2DHWPRF_TLB, Level 3 cache refill, L2D hardware TLB descriptor prefetcher access, event

Counts level 3 cache accesses from L2D hardware TLB descriptor prefetches 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 this Telemetry Specification.

Functional groups

L3_Cache

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

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

0x8151 L3D_CACHE_PRFM, Level 3 data cache software preload, event

Counts level 3 cache accesses generated by software preload or prefetch instructions targeting the level 3 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

0x8156 L3D_CACHE_HWPRF, Level 3 data cache hardware prefetch, event

Counts level 3 cache accesses generated by hardware prefetchers targeting the level 3 cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

`0x81C6` L3D_CACHE_HIT_RD, Level 3 data cache demand hit, read, event

The counter counts each demand read counted by L3D_CACHE_RD that hits in the Level 3 data or unified cache.

Related telemetry artifacts

Metrics

- system_l3_cache_hit_ratio

Metric groups

System_Memory_Effectiveness

Functional groups

L3_Cache

`0x81E6` L3D_CACHE_HIT_RD_FHWPRF, Level 3 data cache demand first hit, read, fetched by hardware prefetcher, event

The counter counts each first hit counted by L3D_CACHE_HIT_RW_FHWPRF that is a demand Memory-read operation.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

`0x8286` L3D_CACHE_PRF, Level 3 data cache, preload or prefetch hit, event

Counts level 3 cache accesses counted by either L3D_CACHE_HWPRF or L3D_CACHE_PRFM.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

L3_Cache

6.8 LL_Cache (LL CACHE) events for C1-Nano

Last Level Cache related events.

Summary of events in LL_Cache:

- Total implemented Common events: 7
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-8: LL_Cache events summary

Code	Mnemonic	Name	Description
0x0032	LL_CACHE	Last level cache access	Counts transactions that were returned from outside the core cluster. This event counts...
0x0036	LL_CACHE_RD	Last level cache access, read	Counts read transactions that were returned from outside the core cluster. This event counts when...
0x0037	LL_CACHE_MISS_RD	Last level cache miss, read	Counts read transactions that were returned from outside the core cluster but missed in the...
0x8157	LL_CACHE_HWPRF	Last level cache hardware prefetch	The counter counts each access counted by LL_CACHE that is due to a hardware prefetch. The...
0x8287	LL_CACHE_PRF	Last level cache, preload or prefetch hit	The counter counts each fetch counted by either LL_CACHE_HWPRF or LL_CACHE_PRFM.
0x8298	LL_CACHE_RW	Last level cache demand access	Counts read and write transactions that were returned from outside the core cluster.
0x8299	LL_CACHE_PRFM	Last level cache software preload	The counter counts each access counted by LL_CACHE that is due to a preload or prefetch...

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x0032 LL_CACHE, Last level cache access, event

Counts transactions that were returned from outside the core cluster. This event counts transactions for external last level cache when the system register IMP_CPUECTLR_EL1.EXTLLC bit is set, otherwise it counts transactions for L3 cache if L2 and L3 caches are present, or for the L2 cache if no L3 is present.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

LL_Cache

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

Counts read transactions that were returned from outside the core cluster. This event counts when the system register IMP_CPUECTLR_EL1.EXTLLC bit is set. 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 when the system register IMP_CPUECTLR_EL1.EXTLLC bit is set. 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

`0x8157` LL_CACHE_HWPRF, Last level cache hardware prefetch, event

The counter counts each access counted by LL_CACHE that is due to a hardware prefetch.

The hardware prefetch is generated by a hardware prefetcher at the Last level cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

LL_Cache

0x8287 LL_CACHE_PRF, Last level cache, preload or prefetch hit, event

The counter counts each fetch counted by either LL_CACHE_HWPRF or LL_CACHE_PRFM.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

LL_Cache

0x8298 LL_CACHE_RW, Last level cache demand access, event

Counts read and write transactions that were returned from outside the core cluster.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

LL_Cache

0x8299 LL_CACHE_PRFM, Last level cache software preload, event

The counter counts each access counted by LL_CACHE that is due to a preload or prefetch instruction.

This includes accesses to the Last level cache due to a refill of another cache caused by a preload or prefetch instruction.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

LL_Cache

6.9 Memory (MEMORY) events for C1-Nano

Memory system related events.

Summary of events in Memory:

- Total implemented Common events: 17
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-9: Memory events summary

Code	Mnemonic	Name	Description
0x0013	MEM_ACCESS	Data memory access	Counts memory accesses issued by the CPU load store unit, where those accesses are issued due to...
0x001A	MEMORY_ERROR	Local memory error	Counts any detected correctable or uncorrectable physical memory errors (ECC or parity) in...
0x0031	REMOTE_ACCESS	Access to another socket in a multi-socket system	Counts accesses to another chip, which is implemented as a different CMN mesh in the system. If...
0x0038	REMOTE_ACCESS_RD	Access to another socket in a multi-socket system, read	Counts read accesses to another chip, which is implemented as a different CMN mesh in the system....
0x0066	MEM_ACCESS_RD	Data memory access, read	Counts memory accesses issued by the CPU due to load operations. The event counts any memory load...
0x0067	MEM_ACCESS_WR	Data memory access, write	Counts memory accesses issued by the CPU due to store operations. The event counts any memory...
0x4020	LDST_ALIGN_LAT	Access with additional latency from alignment	Counts the number of memory read and write accesses in a cycle that incurred additional latency,...
0x4021	LD_ALIGN_LAT	Load with additional latency from alignment	Counts the number of memory read accesses in a cycle that incurred additional latency, due to the...
0x4022	ST_ALIGN_LAT	Store with additional latency from alignment	Counts the number of memory write access in a cycle that incurred additional latency, due to the...
0x4024	MEM_ACCESS_CHECKED	Checked data memory access	Counts the number of memory read and write accesses in a cycle that that are tag checked by the...
0x4025	MEM_ACCESS_CHECKED_RD	Checked data memory access, read	Counts the number of memory read accesses in a cycle that are tag checked by the Memory Tagging...
0x4026	MEM_ACCESS_CHECKED_WR	Checked data memory access, write	Counts the number of memory write accesses in a cycle that is tag checked by the Memory Tagging...
0x8120	INST_FETCH_PERCYC	Event in progress, INST FETCH	Counts number of instruction fetches outstanding per cycle, which will provide an average latency...
0x8121	MEM_ACCESS_RD_PERCYC	Event in progress, MEM ACCESS RD	Counts the number of outstanding loads or memory read accesses per cycle.
0x8124	INST_FETCH	Instruction memory access	Counts instruction memory accesses in a given cycle.
0x82A0	MEM_ACCESS_RW	Data memory access, demand access	Counts memory accesses issued by the CPU due to load and store operations.
0x82A1	INST_FETCH_RD	Instruction memory access, demand fetch	Counts instruction memory accesses in a given cycle that are not prefetches.

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

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

Metrics

- load_average_latency

Metric groups

Average_Latency

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 this Telemetry 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 this Telemetry Specification.

Functional groups

Memory

`0x0038` REMOTE_ACCESS_RD, Access to another socket in a multi-socket system, read, event

Counts read 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 this Telemetry 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 this Telemetry 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 this Telemetry Specification.

Functional groups

Memory

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

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Memory

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

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Memory

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

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Memory

0x4024 MEM_ACCESS_CHECKED, Checked data memory access, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Memory

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

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Memory

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

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Memory

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

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

Related telemetry artifacts

Metrics

- instruction_fetch_average_latency

Metric groups

Average_Latency

Functional groups

Memory

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

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

Related telemetry artifacts

Metrics

- load_average_latency

Metric groups

Average_Latency

Functional groups

Memory

`0x8124` INST_FETCH, Instruction memory access, event

Counts instruction memory accesses in a given cycle.

Related telemetry artifacts

Metrics

- instruction_fetch_average_latency

Metric groups

Average_Latency

Functional groups

Memory

`0x82A0` MEM_ACCESS_RW, Data memory access, demand access, event

Counts memory accesses issued by the CPU due to load and store operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Memory

`0x82A1` INST_FETCH_RD, Instruction memory access, demand fetch, event

Counts instruction memory accesses in a given cycle that are not prefetches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Memory

6.10 Retired (RETIRED) events for C1-Nano

Retired instruction and operation events.

Summary of events in Retired:

- Total implemented Common events: 44
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-10: Retired events summary

Code	Mnemonic	Name	Description
0x0000	SW_INCR	Instruction architecturally executed, Condition code check pass, software increment	Counts software writes to the PMSWINC_EL0 (software PMU increment) register. The PMSWINC_EL0...
0x0006	LD_RETIRED	Instruction architecturally executed, Condition code check pass, load	Counts load instructions that have been architecturally executed.
0x0007	ST_RETIRED	Instruction architecturally executed, Condition code check pass, store	Counts store operations that have been architecturally executed.
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...
0x000C	PC_WRITE_RETIRED	Instruction architecturally executed, Condition code check pass, Software change of the PC	Counts branch instructions that caused a change of Program Counter, which effectively causes a...
0x000D	BR_IMMED_RETIRED	Branch instruction architecturally executed, immediate	Counts architecturally executed direct branches.
0x000E	BR_RETURN_RETIRED	Branch instruction architecturally executed, procedure return, taken	Counts architecturally executed procedure returns, both correctly predicted as well as mispredicted.
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...

Code	Mnemonic	Name	Description
0x0022	BR_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted	Counts branches counted by BR_RETIRED which were mispredicted and caused a pipeline flush.
0x003A	OP_RETIRED	Micro-operation architecturally executed	Counts micro-operations that are architecturally executed. This is a count of number of...
0x8000	SIMD_INST_RETIRED	Instruction architecturally executed, SIMD	Counts executed operations that are SIMD or SVE vector operations or Advanced SIMD non-scalar...
0x8107	BR_SKIP_RETIRED	Branch instruction architecturally executed, not taken	Counts architecturally executed branches that were not taken.
0x8108	BR_IMMED_TAKEN_RETIRED	Branch instruction architecturally executed, immediate, taken	Counts architecturally executed immediate branches that were conditional and taken.
0x810C	BR_INDNR_TAKEN_RETIRED	Branch instruction architecturally executed, indirect excluding procedure return, taken	Counts architecturally executed indirect branches excluding procedure returns that were taken.
0x8110	BR_IMMED_PRED_RETIRED	Branch instruction architecturally executed, predicted immediate	Counts architecturally executed direct branches that were correctly predicted.
0x8111	BR_IMMED_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted immediate	Counts architecturally executed direct branches that were mispredicted and caused a pipeline flush.
0x8112	BR_IND_PRED_RETIRED	Branch instruction architecturally executed, predicted indirect	Counts architecturally executed indirect branches including procedure returns that were correctly...
0x8113	BR_IND_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted indirect	Counts architecturally executed indirect branches including procedure returns that were...
0x8114	BR_RETURN_PRED_RETIRED	Branch instruction architecturally executed, predicted procedure return	Counts architecturally executed procedure returns that were correctly predicted.
0x8115	BR_RETURN_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted procedure return	Counts architecturally executed procedure returns that were mispredicted and caused a pipeline...
0x8116	BR_INDNR_PRED_RETIRED	Branch instruction architecturally executed, predicted indirect excluding procedure return	Counts architecturally executed indirect branches excluding procedure returns that were correctly...
0x8117	BR_INDNR_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted indirect excluding procedure return	Counts architecturally executed indirect branches excluding procedure returns that were...
0x8118	BR_TAKEN_PRED_RETIRED	Branch instruction architecturally executed, predicted branch, taken	Counts architecturally executed branches that were taken and were correctly predicted.
0x8119	BR_TAKEN_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted branch, taken	Counts architecturally executed branches that were taken and were mispredicted causing a pipeline...
0x811A	BR_SKIP_PRED_RETIRED	Branch instruction architecturally executed, predicted branch, not taken	Counts architecturally executed branches that were not taken and were correctly predicted.
0x811B	BR_SKIP_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted branch, not taken	Counts architecturally executed branches that were not taken and were mispredicted causing a...

Code	Mnemonic	Name	Description
0x811C	BR_PRED_RETIRED	Branch instruction architecturally executed, predicted branch	Counts branch instructions counted by BR_RETIRED which were correctly predicted.
0x811D	BR_IND_RETIRED	Instruction architecturally executed, indirect branch	Counts architecturally executed indirect branches including procedure returns.
0x8179	BRNL_INDNR_TAKEN_RETIRED	Branch instruction architecturally executed, indirect branch without link excluding procedure return, taken	Counts architecturally executed indirect branch without link instructions, excluding return...
0x817A	BL_TAKEN_RETIRED	Branch instruction architecturally executed, branch with link, taken	Counts architecturally executed branch with link instructions that were taken.
0x817B	BRNL_TAKEN_RETIRED	Branch instruction architecturally executed, branch without link, taken	Counts architecturally executed branch without link instructions that were taken.
0x817C	BL_IND_TAKEN_RETIRED	Branch instruction architecturally executed, indirect branch with link, taken	Counts architecturally executed indirect branch with link instructions that were taken.
0x817D	BRNL_IND_TAKEN_RETIRED	Branch instruction architecturally executed, indirect branch without link, taken	Counts architecturally executed indirect branch without link instructions that were taken.
0x817E	BL_IMMED_TAKEN_RETIRED	Branch instruction architecturally executed, direct branch with link, taken	Counts architecturally executed direct branch with link instructions that were taken.
0x817F	BRNL_IMMED_TAKEN_RETIRED	Branch instruction architecturally executed, direct branch without link, taken	Counts architecturally executed direct branch without link instructions that were taken.
0x8180	BR_UNCOND_RETIRED	Branch instruction architecturally executed, unconditional branch	Counts architecturally executed unconditional branch instructions.
0x8181	BR_COND_RETIRED	Branch instruction architecturally executed, conditional branch	Counts architecturally executed conditional branch instructions.
0x8182	BR_COND_TAKEN_RETIRED	Branch instruction architecturally executed, conditional branch, taken	Counts architecturally executed conditional branch instructions that were taken.
0x8183	BR_HINT_COND_RETIRED	Branch instruction architecturally executed, hinted conditional.	Counts architecturally executed hinted conditional branches.
0x8184	BR_HINT_COND_PRED_RETIRED	Branch instruction architecturally executed, predicted hinted conditional.	Counts architecturally executed hinted conditional branches that were correctly predicted.
0x8185	BR_HINT_COND_MIS_PRED_RETIRED	Branch instruction architecturally executed, mispredicted hinted conditional	Counts architecturally executed hinted conditional branches that were mispredicted and caused a...
0x8186	UOP_RETIRED	Other micro-operation architecturally executed	Counts micro-operations executed.

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

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

Counts software writes to the PMSWINC_EL0 (software PMU increment) register. The PMSWINC_EL0 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_EL0 register each time the data structure is accessed).

To use the PMSWINC_EL0 register and event, developers must insert instructions that write to the PMSWINC_EL0 register into the source code.

Since the SW_INCR event records writes to the PMSWINC_EL0 register, there is no need to do a read/increment/write sequence to the PMSWINC_EL0 register.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x0006 LD_RETIRED, Instruction architecturally executed, Condition code check pass, load, event

Counts load instructions that have been architecturally executed.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x0007 ST_RETIRED, Instruction architecturally executed, Condition code check pass, store, event

Counts store operations that have been architecturally executed.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x0008 INST_RETIRED, Instruction architecturally executed, event

Counts instructions that have been architecturally executed.

Related telemetry artifacts

Metrics

- ipc
- branch_mpki in Branch_Effectiveness
- branch_mpki in MPKI
- itlb_mpki in ITLB_Effectiveness
- itlb_mpki in MPKI

- l1i_tlb_mpki in ITLB_Effectiveness
- l1i_tlb_mpki in MPKI
- dtlb_mpki in DTLB_Effectiveness
- dtlb_mpki in MPKI
- l1d_tlb_mpki in DTLB_Effectiveness
- l1d_tlb_mpki in MPKI
- l2_tlb_mpki in DTLB_Effectiveness
- l2_tlb_mpki in ITLB_Effectiveness
- l2_tlb_mpki in MPKI
- l1i_cache_mpki in L1I_Cache_Effectiveness
- l1i_cache_mpki in MPKI
- l1d_cache_mpki in L1D_Cache_Effectiveness
- l1d_cache_mpki in MPKI
- l2i_cache_mpki in L2I_Cache_Effectiveness
- l2i_cache_mpki in MPKI
- l2d_cache_mpki in L2D_Cache_Effectiveness
- l2d_cache_mpki in MPKI
- l3_cache_mpki in L3_Cache_Effectiveness
- l3_cache_mpki in MPKI
- ll_cache_read_mpki in LL_Cache_Effectiveness
- ll_cache_read_mpki in MPKI

Metric groups

Branch_Effectiveness
DTLB_Effectiveness
General
ITLB_Effectiveness
L1D_Cache_Effectiveness
L1I_Cache_Effectiveness
L2D_Cache_Effectiveness
L2I_Cache_Effectiveness
L3_Cache_Effectiveness
LL_Cache_Effectiveness
MPKI

Functional groups

Retired

0x000B CID_WRITE_RETIRED, Instruction architecturally executed, Condition code check pass, write to CONTEXTIDR, event

Counts architecturally executed writes to the CONTEXTIDR 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 this Telemetry Specification.

Functional groups

Retired

0x000C PC_WRITE_RETIRED, Instruction architecturally executed, Condition code check pass, Software change of the PC, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x000D BR_IMMED_RETIRED, Branch instruction architecturally executed, immediate, event

Counts architecturally executed direct branches.

Related telemetry artifacts

Metrics

- branch_direct_ratio

Metric groups

Branch_Effectiveness

Functional groups

Retired

0x000E BR_RETURN_RETIRED, Branch instruction architecturally executed, procedure return, taken, event

Counts architecturally executed procedure returns, both correctly predicted as well as mispredicted.

Related telemetry artifacts

Metrics

- branch_return_ratio

Metric groups

Branch_Effectiveness

Functional groups

Retired

0x001C TTBR_WRITE_RETIRED, Instruction architecturally executed, Condition code check pass, write to TTBR, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry 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_direct_ratio
- branch_indirect_ratio
- branch_return_ratio
- branch_misprediction_ratio in Branch_Effectiveness
- branch_misprediction_ratio in Miss_Ratio

Metric groups

Branch_Effectiveness

Miss_Ratio

Functional groups

Retired

0x0022 BR_MIS_PRED_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

0x003A OP_RETIRED, Micro-operation architecturally executed, event

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

Related telemetry artifacts

Metrics

- retiring
- bad_speculation

Metric groups

Topdown_L1

Functional groups

Retired

0x8000 SIMD_INST_RETIRED, Instruction architecturally executed, SIMD, event

Counts executed operations that are SIMD or SVE vector operations or Advanced SIMD non-scalar operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x8107 BR_SKIP_RETIRED, Branch instruction architecturally executed, not taken, event

Counts architecturally executed branches that were not taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

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

Counts architecturally executed immediate branches that were conditional and taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x810C` BR_INDNR_TAKEN_RETIRED, Branch instruction architecturally executed, indirect excluding procedure return, taken, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8110` BR_IMMED_PRED_RETIRED, Branch instruction architecturally executed, predicted immediate, event

Counts architecturally executed direct branches that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8111` BR_IMMED_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted immediate, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8112` BR_IND_PRED_RETIRED, Branch instruction architecturally executed, predicted indirect, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8113` BR_IND_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted indirect, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8114` BR_RETURN_PRED_RETIRED, Branch instruction architecturally executed, predicted procedure return, event

Counts architecturally executed procedure returns that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8115` BR_RETURN_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted procedure return, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

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

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x8117 BR_INDNR_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted indirect excluding procedure return, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x8118 BR_TAKEN_PRED_RETIRED, Branch instruction architecturally executed, predicted branch, taken, event

Counts architecturally executed branches that were taken and were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x8119 BR_TAKEN_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted branch, taken, event

Counts architecturally executed branches that were taken and were mispredicted causing a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x811A BR_SKIP_PRED_RETIRED, Branch instruction architecturally executed, predicted branch, not taken, event

Counts architecturally executed branches that were not taken and were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x811B BR_SKIP_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted branch, not taken, event

Counts architecturally executed branches that were not taken and were mispredicted causing a pipeline flush.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x811C BR_PRED_RETIRED, Branch instruction architecturally executed, predicted branch, event

Counts branch instructions counted by BR_RETIRED which were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x811D BR_IND_RETIRED, Instruction architecturally executed, indirect branch, event

Counts architecturally executed indirect branches including procedure returns.

Related telemetry artifacts

Metrics

- branch_indirect_ratio

Metric groups

Branch_Effectiveness

Functional groups

Retired

0x8179 BRNL_INDNR_TAKEN_RETIRED, Branch instruction architecturally executed, indirect branch without link excluding procedure return, taken, event

Counts architecturally executed indirect branch without link instructions, excluding return instructions, that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x817A BL_TAKEN_RETIRED, Branch instruction architecturally executed, branch with link, taken, event

Counts architecturally executed branch with link instructions that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x817B BRNL_TAKEN_RETIRED, Branch instruction architecturally executed, branch without link, taken, event

Counts architecturally executed branch without link instructions that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x817C BL_IND_TAKEN_RETIRED, Branch instruction architecturally executed, indirect branch with link, taken, event

Counts architecturally executed indirect branch with link instructions that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x817D BRNL_IND_TAKEN_RETIRED, Branch instruction architecturally executed, indirect branch without link, taken, event

Counts architecturally executed indirect branch without link instructions that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x817E BL_IMMED_TAKEN_RETIRED, Branch instruction architecturally executed, direct branch with link, taken, event

Counts architecturally executed direct branch with link instructions that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

0x817F BRNL_IMMED_TAKEN_RETIRED, Branch instruction architecturally executed, direct branch without link, taken, event

Counts architecturally executed direct branch without link instructions that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8180` BR_UNCOND_RETIRED, Branch instruction architecturally executed, unconditional branch, event

Counts architecturally executed unconditional branch instructions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8181` BR_COND_RETIRED, Branch instruction architecturally executed, conditional branch, event

Counts architecturally executed conditional branch instructions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8182` BR_COND_TAKEN_RETIRED, Branch instruction architecturally executed, conditional branch, taken, event

Counts architecturally executed conditional branch instructions that were taken.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8183` BR_HINT_COND_RETIRED, Branch instruction architecturally executed, hinted conditional., event

Counts architecturally executed hinted conditional branches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8184` BR_HINT_COND_PRED_RETIRED, Branch instruction architecturally executed, predicted hinted conditional., event

Counts architecturally executed hinted conditional branches that were correctly predicted.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8185` BR_HINT_COND_MIS_PRED_RETIRED, Branch instruction architecturally executed, mispredicted hinted conditional, event

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

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

`0x8186` UOP_RETIRED, Other micro-operation architecturally executed, event

Counts micro-operations executed.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Retired

6.11 Spec_Operation (SPEC OPERATION) events for C1-Nano

Speculatively executed operations related events.

Summary of events in Spec_Operation:

- Total implemented Common events: 48
- Total Implemented Product ImpDef events: 3
- PMU Only events : 3
- ETE Only events : 0

Table 6-11: 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 speculatively executed predictable branches.
0x001B	INST_SPEC	Operation speculatively executed	Counts operations that have been speculatively executed.
0x003B	OP_SPEC	Micro-operation speculatively executed	Counts micro-operations speculatively executed. This is the count of the number of...
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.
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)...
0x0072	LDST_SPEC	Operation speculatively executed, load or store	Counts load and store operations that have been speculatively executed. Operations that perform a...
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 data processing	The counter counts each operation counted by INST_SPEC that is an Advanced SIMD...
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	DSB_SPEC	Barrier speculatively executed, DSB	Counts DSB operations that are speculatively issued to Load/Store unit in the CPU.
0x007E	DMB_SPEC	Barrier speculatively executed, DMB	Counts DMB operations that are speculatively issued to the Load/Store unit in the CPU. This event...
0x007F	CSDB_SPEC	Barrier speculatively executed, CSDB	Counts CDSB operations that are speculatively issued to the Load/Store unit in the CPU. This...
0x0090	RC_LD_SPEC	Release consistency operation speculatively executed, Load-Acquire	Counts any load acquire operations that are speculatively executed. For example: LDAR, LDARH, LDARB
0x0091	RC_ST_SPEC	Release consistency operation speculatively executed, Store-Release	Counts any store release operations that are speculatively executed. For example: STLR, STLRH,...
0x3218	OP_CME_ISSUE	SME operation issued	The counter counts each operation counted by OP_ISSUE that was issued to a streaming mode compute...
0x3219	SSVE_INST_SPEC	Operation speculatively executed, Streaming SVE, including load and store	The counter counts each instruction counted by SVE_INST_SPEC when the CPU executes in Streaming...
0x321a	SSVE_SPEC	Operation speculatively executed, Streaming SVE	The counter counts each operation counted by SVE_SPEC specifically in Streaming mode.
0x8004	SIMD_INST_SPEC	Operation speculatively executed, SIMD	Counts speculatively executed operations that are SIMD or SVE vector operations or Advanced SIMD...
0x8005	ASE_INST_SPEC	Operation speculatively executed, Advanced SIMD	The counter counts each Speculatively executed operation due to an A64 Advanced...
0x8006	SVE_INST_SPEC	Operation speculatively executed, SVE, including load and store	The counter counts each Speculatively executed operation due to an SVE instruction. It is...
0x8056	SVE_SPEC	Operation speculatively executed, SVE	The counter counts each operation counted by INST_SPEC that is a scalable vector data processing...
0x8057	ASE_SVE_SPEC	Operation speculatively executed, Advanced SIMD or SVE data processing	The counter counts each operation counted by INST_SPEC that is an Advanced SIMD or scalable...
0x8080	SVE_LDST_SPEC	Operation speculatively executed, SVE load, store, or prefetch	The counter counts each Speculatively executed load, store, or prefetch operation counted by...
0x8081	SVE_LD_SPEC	Operation speculatively executed, SVE load	The counter counts each Speculatively executed operation that reads from memory due to an...
0x8082	SVE_ST_SPEC	Operation speculatively executed, SVE store	The counter counts each Speculatively executed operation that writes to memory due to an...
0x8083	SVE_PRF_SPEC	Operation speculatively executed, SVE prefetch	The counter counts each Speculatively executed prefetch operation due to any of the following...
0x8170	CAS_NEAR_FAIL	Atomic memory Operation speculatively executed, Compare and Swap fail	This event counts compare and swap instructions that executed locally to the PE and did not...
0x8171	CAS_NEAR_PASS	Atomic memory Operation speculatively executed, Compare and Swap pass	This event counts compare and swap instructions that executed locally to the PE and updated the...
0x8172	CAS_NEAR_SPEC	Atomic memory Operation speculatively executed, Compare and Swap near	This event counts compare and swap instructions that executed locally to the PE.
0x8173	CAS_FAR_SPEC	Atomic memory Operation speculatively executed, Compare and Swap far	This event counts compare and swap instructions that did not execute locally to the PE.
0x8174	CAS_SPEC	Atomic memory Operation speculatively executed, Compare and Swap	This event counts the total compare and swap instructions that were executed.

Code	Mnemonic	Name	Description
0x8175	LSE_LD_SPEC	Atomic memory Operation speculatively executed, load	This event counts the total atomic memory instructions that return a value that were...
0x8176	LSE_ST_SPEC	Atomic memory Operation speculatively executed, store	This event counts the total atomic memory instructions that do not return a value that were...
0x8177	LSE_LDST_SPEC	Atomic memory Operation speculatively executed, load or store	This event counts the total atomic memory instructions that were speculatively executed.
0x835D	SE_SPEC	Operation speculatively executed, Advanced SIMD, SVE or SME data processing	The counter counts each operation counted by INST_SPEC that is an Advanced SIMD, scalable...
0x835E	SME_INST_SPEC	Operation speculatively executed, SME	The counter counts each speculatively executed operation counted by SE_INST_SPEC that is...
0x835F	SE_INST_SPEC	Operation speculatively executed, Advanced SIMD, SVE, SME	The counter counts each speculatively executed operation counted by INST_SPEC that is classified...

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

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 this Telemetry Specification.

Functional groups

Spec_Operation

0x0012 BR_PRED, Predictable branch instruction speculatively executed, event

Counts speculatively executed predictable branches.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x001B INST_SPEC, Operation speculatively executed, event

Counts operations that have been speculatively executed.

Related telemetry artifacts

Metrics

- load_store_percentage
- load_percentage
- store_percentage
- integer_dp_percentage

- simd_percentage
- scalar_fp_percentage
- branch_percentage
- crypto_percentage
- sve_percentage
- sve_predicate_percentage
- fp16_percentage
- fp32_percentage
- fp64_percentage
- sme_percentage

Metric groups

FP_Precision_Mix
Operation_Mix
SVE_Effectiveness

Functional groups

Spec_Operation

0x003B OP_SPEC, Micro-operation speculatively executed, event

Counts micro-operations speculatively executed. This is the count of the number of micro-operations dispatched in a cycle.

Related telemetry artifacts

Metrics

- retiring
- bad_speculation

Metric groups

Topdown_L1

Functional groups

Spec_Operation

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 this Telemetry 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 this Telemetry 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 this Telemetry Specification.

Functional groups

Spec_Operation

0x006C LDREX_SPEC, Exclusive operation speculatively executed, Load-Exclusive, event

Counts Load-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 this Telemetry 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 this Telemetry 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 this Telemetry 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 this Telemetry 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_ls_percentage
- 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_ls_percentage
- store_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x0072 LDST_SPEC, Operation speculatively executed, load or store, event

Counts load and store operations that have been speculatively executed. Operations that perform a load and a store operation are counted only once

Related telemetry artifacts

Metrics

- load_store_percentage
- load_ls_percentage
- store_ls_percentage
- lse_atomics_ratio

Metric groups

Atomics_Effectiveness
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 data processing, event

The counter counts each operation counted by INST_SPEC that is an Advanced SIMD data-processing operation. It does not count SVE operations counted by SVE_SPEC, or SME operations counted by SME_SPEC.

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 this Telemetry 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 this Telemetry 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. For example: 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 this Telemetry 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 this Telemetry 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 this Telemetry Specification.

Functional groups

Spec_Operation

0x007F CSDB_SPEC, Barrier speculatively executed, CSDB, event

Counts CDSB 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 this Telemetry 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. For example: LDAR, LDARH, LDARB

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x0091 RC_ST_SPEC, Release consistency operation speculatively executed, Store-Release, event

Counts any store release operations that are speculatively executed. For example: STLR, STLRH, STLRB'

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x3218 OP_CME_ISSUE, SME operation issued, event

The counter counts each operation counted by OP_ISSUE that was issued to a streaming mode compute unit.

The definition of which operations are issued to an SME2 unit is **IMPLEMENTATION DEFINED**. The maximum value by which the counter could increment by in a single cycle is **IMPLEMENTATION DEFINED**.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

`0x3219` SSVE_INST_SPEC, Operation speculatively executed, Streaming SVE, including load and store, event

The counter counts each instruction counted by SVE_INST_SPEC when the CPU executes in Streaming mode.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

`0x321a` SSVE_SPEC, Operation speculatively executed, Streaming SVE, event

The counter counts each operation counted by SVE_SPEC specifically in Streaming mode.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

`0x8004` SIMD_INST_SPEC, Operation speculatively executed, SIMD, event

Counts speculatively executed operations that are SIMD or SVE vector operations or Advanced SIMD non-scalar operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

`0x8005` ASE_INST_SPEC, Operation speculatively executed, Advanced SIMD, event

The counter counts each Speculatively executed operation due to an A64 Advanced SIMD instruction.

It is **IMPLEMENTATION DEFINED** whether the counter counts operations due to Advanced SIMD scalar instructions.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x8006 SVE_INST_SPEC, Operation speculatively executed, SVE, including load and store, event

The counter counts each Speculatively executed operation due to an SVE instruction.

It is **IMPLEMENTATION DEFINED** whether the counter counts operations due to non-SIMD SVE instructions.

Instructions classified as SME instructions and counted by SME_INST_SPEC are not counted by this event.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation
SVE

0x8056 SVE_SPEC, Operation speculatively executed, SVE, event

The counter counts each operation counted by INST_SPEC that is a scalable vector data processing operation.

It does not count:

- SME operations counted by SME_SPEC.
- Neon operation counted by ASE_SPEC
- Load/store operations

Related telemetry artifacts

Metrics

- sve_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x8057 ASE_SVE_SPEC, Operation speculatively executed, Advanced SIMD or SVE data processing, event

The counter counts each operation counted by INST_SPEC that is an Advanced SIMD or scalable vector data processing operation.

It does not count:

- SME operations counted by SME_SPEC.
- Load/store operations

See ASE_SPEC and SVE_SPEC for these classifications.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x8080 SVE_LDST_SPEC, Operation speculatively executed, SVE load, store, or prefetch, event

The counter counts each Speculatively executed load, store, or prefetch operation counted by any of SVE_LD_SPEC, SVE_PRF_SPEC, or SVE_ST_SPEC.

This includes Load/Store operations to Z register but does not count Load/Store operations to ZT and ZA registers.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x8081 SVE_LD_SPEC, Operation speculatively executed, SVE load, event

The counter counts each Speculatively executed operation that reads from memory due to an SVE load instruction.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x8082 SVE_ST_SPEC, Operation speculatively executed, SVE store, event

The counter counts each Speculatively executed operation that writes to memory due to an SVE store instruction.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x8083 SVE_PRF_SPEC, Operation speculatively executed, SVE prefetch, event

The counter counts each Speculatively executed prefetch operation due to any of the following A64 instructions:

- SVE: PRFB, PRFD, PRFH, or PRFW.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x8170 CAS_NEAR_FAIL, Atomic memory Operation speculatively executed, Compare and Swap fail, event

This event counts compare and swap instructions that executed locally to the PE and did not update the location accessed.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x8171 CAS_NEAR_PASS, Atomic memory Operation speculatively executed, Compare and Swap pass, event

This event counts compare and swap instructions that executed locally to the PE and updated the location accessed.

Related telemetry artifacts

Metrics

- cas_near_pass_ratio

Metric groups

Atomics_Effectiveness

Functional groups

Spec_Operation

0x8172 CAS_NEAR_SPEC, Atomic memory Operation speculatively executed, Compare and Swap near, event

This event counts compare and swap instructions that executed locally to the PE.

Related telemetry artifacts

Metrics

- cas_near_ratio
- cas_far_ratio
- cas_near_pass_ratio

Metric groups

Atomics_Effectiveness

Functional groups

Spec_Operation

`0x8173` CAS_FAR_SPEC, Atomic memory Operation speculatively executed, Compare and Swap far, event

This event counts compare and swap instructions that did not execute locally to the PE.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

`0x8174` CAS_SPEC, Atomic memory Operation speculatively executed, Compare and Swap, event

This event counts the total compare and swap instructions that were executed.

Related telemetry artifacts

Metrics

- cas_near_ratio
- cas_far_ratio

Metric groups

Atomics_Effectiveness

Functional groups

Spec_Operation

`0x8175` LSE_LD_SPEC, Atomic memory Operation speculatively executed, load, event

This event counts the total atomic memory instructions that return a value that were speculatively executed.

Related telemetry artifacts

Metrics

- lse_load_ratio

Metric groups

Atomics_Effectiveness

Functional groups

Spec_Operation

0x8176 LSE_ST_SPEC, Atomic memory Operation speculatively executed, store, event

This event counts the total atomic memory instructions that do not return a value that were speculatively executed.

Related telemetry artifacts

Metrics

- lse_store_ratio

Metric groups

Atomics_Effectiveness

Functional groups

Spec_Operation

0x8177 LSE_LDST_SPEC, Atomic memory Operation speculatively executed, load or store, event

This event counts the total atomic memory instructions that were speculatively executed.

Related telemetry artifacts

Metrics

- lse_atomics_ratio
- lse_load_ratio
- lse_store_ratio

Metric groups

Atomics_Effectiveness

Functional groups

Spec_Operation

0x835D SE_SPEC, Operation speculatively executed, Advanced SIMD, SVE or SME data processing, event

The counter counts each operation counted by INST_SPEC that is an Advanced SIMD, scalable vector extension, or scalable matrix extension data-processing operation.

See ASE_SVE_SPEC and SME_SPEC for these classifications.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

0x835E SME_INST_SPEC, Operation speculatively executed, SME, event

The counter counts each speculatively executed operation counted by SE_INST_SPEC that is classified as an SME operation.

Operations due to the following instructions are counted as SME operations:

- Data-processing operations involving the ZA and ZT registers.
- Load and store operations involving the ZA and ZT registers.

Operations due to instructions added by FEAT_SME which involve the SVE registers but do not involve any ZA or ZT registers are counted as SVE data-processing operations.

Related telemetry artifacts

Metrics

- sme_percentage

Metric groups

Operation_Mix

Functional groups

Spec_Operation

0x835F SE_INST_SPEC, Operation speculatively executed, Advanced SIMD, SVE, SME, event

The counter counts each speculatively executed operation counted by INST_SPEC that is classified as an Advanced SIMD, scalable vector extension, or scalable matrix extension operation.

See ASE_SVE_INST_SPEC and SME_INST_SPEC for these classifications

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

6.12 Stall (STALL) events for C1-Nano

Stall related events.

Summary of events in Stall:

- Total implemented Common events: 23
- Total Implemented Product ImpDef events: 25
- PMU Only events : 25
- ETE Only events : 0

Table 6-12: Stall events summary

Code	Mnemonic	Name	Description
0x0023	STALL_FRONTEND	No operation sent for execution due to the frontend	Counts cycles when frontend could not send any micro-operations to the dispatch logic because of...
0x0024	STALL_BACKEND	No operation sent for execution due to the backend	Counts cycles whenever the dispatch logic is unable to send any micro-operations to the backend...
0x003C	STALL	No operation sent for execution	Counts cycles when no operations are sent to the dispatch unit from the frontend or from the...
0x003D	STALL_SLOT_BACKEND	No operation sent for execution on a Slot due to the backend	Counts slots per cycle in which an operation is available to the dispatch unit, but the operation...
0x003E	STALL_SLOT_FRONTEND	No operation sent for execution on a Slot due to the frontend	Counts slots per cycle in which no operations are sent to the dispatch logic from the frontend...
0x003F	STALL_SLOT	No operation sent for execution on a Slot	Counts slots per cycle in which no operations are sent to the dispatch logic from the frontend or...
0x00EE	IMP_STALL_SLOT_BACKEND_ILOCK	Backend slot stall, input dependency	Counts slots per cycle in which an operation is available to the dispatch unit, and the operation...
0x00e6	IMP_STALL_BACKEND_ILOCK_VPU	Backend stall, input dependency from VPU instruction	Counts cycles when the backend is stalled because of an instruction interlock, and the source of...
0x00ef	IMP_STALL_BACKEND_BUSY_VPU_ARB	Backend stall, busy VPU arbitration	Counts cycles when the backend could not accept any micro-operations because of a lack of...
0x00f1	IMP_STALL_BACKEND_BUSY_LS	Backend stall, busy LS unit	Counts cycles when the backend could not accept any micro-operations because of a lack of...
0x00f2	IMP_STALL_BACKEND_ILOCK_LS	Backend stall, input dependency from memory instruction	Counts cycles when the backend is stalled because of an instruction interlock, and the source of...
0x00f3	IMP_STALL_BACKEND_ILOCK_LS_INTO_AGU	Backend stall, memory instruction input dependency from memory instruction (pointer chase)	Counts cycles when the backend is stalled because of an instruction interlock, the source of at...
0x00f4	IMP_STALL_SLOT_BACKEND_PORT_CONTENTION	Backend slot stall, port contention	Counts slots per cycle in which an operation is available to the dispatch unit, and the operation...
0x3200	STALL_BACKEND_BUSY_CME	Backend stall cycles, SME2 unit busy	The counter counts each PE cycle counted by STALL_BACKEND_CPUBOUND when the PEs backend was...
0x3201	STALL_BACKEND_BUSY_CMEBOUND	Backend stall cycles, SME2 unit backpressure	The counter counts each CPU cycle counted by STALL_BACKEND_BUSY_CME when the SME2 unit causes...

Code	Mnemonic	Name	Description
0x3202	STALL_BACKEND_BUSY_CME_ARB	Backend stall cycles, SME2 unit arbitration	The counter counts each CPU cycle counted by STALL_BACKEND_BUSY_CME when oldest SME instruction...
0x3203	STALL_BACKEND_BUSY_CME_CPUBOUND	Backend stall cycles, SME2 unit stalled by CPU	The counter counts each PE cycle counted by STALL_BACKEND_BUSY_CME when not counted by...
0x3204	STALL_BACKEND_ILOCK_FROM_CME	Backend stall, input dependency from SME2 unit	The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK when a CPU instruction waits for...
0x3205	STALL_BACKEND_ILOCK_FROM_CME_GPR	Backend stall, input dependency from SME2 unit general purpose register	The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_FROM_CME when dependency is on a...
0x3206	STALL_BACKEND_ILOCK_FROM_CME_PRED	Backend stall, input dependency from SME2 unit predicate register	The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_FROM_CME when dependency is on a...
0x3207	STALL_BACKEND_ILOCK_FROM_CME_FLAGS	Backend stall, input dependency from SME2 unit flag register	The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_FROM_CME when dependency is...
0x3208	STALL_BACKEND_ILOCK_TO_CME	Backend stall, SME2 unit input dependency from CPU	The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK when oldest SME instruction is...
0x3209	STALL_BACKEND_ILOCK_TO_CME_GPR	Backend stall, SME2 unit input dependency from CPU on general purpose register	The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_TO_CME when oldest SME...
0x320a	STALL_BACKEND_ILOCK_TO_CME_PRED	Backend stall, SME2 unit input dependency from CPU predicate register	The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_TO_CME when operations when...
0x320b	STALL_BACKEND_ILOCK_TO_CME_FLAGS	Backend stall, SME2 unit input dependency from CPU on flags	The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_TO_CME when operations when...
0x320c	STALL_BACKEND_MEM_CME_LSRT_FULL	Backend stall cycles, SME2 unit stalled on LSRT full	The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when at least one Streaming...
0x320d	STALL_BACKEND_MEM_CME_BARRIER	Backend stall cycles, barrier stalled by SME2 unit	The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when a barrier instruction is...
0x320e	STALL_BACKEND_MEM_CME_HZ_ON_CPU	Backend stall cycles, SME2 unit stalled by CPU memory hazard	The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when at least one Streaming...

Code	Mnemonic	Name	Description
0x320f	STALL_BACKEND_MEM_CPU_HZ_ON_CME	Backend stall cycles, CPU stalled by SME2 unit memory hazard	The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when at least one CPU...
0x3210	STALL_BACKEND_MEM_CME	Backend stall cycles, SME2 unit stalled on memory hazard or LSRT full	The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when at least one Streaming...
0x4005	STALL_BACKEND_MEM	Memory stall cycles	Counts cycles when the backend is stalled, and there is a demand data miss in the last level core...
0x8158	STALL_FRONTEND_MEMBOUND	Frontend stall cycles, memory bound	Counts cycles when the frontend could not send any micro-operations to the dispatch logic stage...
0x8159	STALL_FRONTEND_L1I	Frontend stall cycles, level 1 instruction cache	Counts cycles when the frontend is stalled because there is an instruction fetch request pending...
0x815B	STALL_FRONTEND_MEM	Frontend stall cycles, last level PE cache or memory	Counts cycles when the frontend is stalled because there is an instruction fetch request pending...
0x815C	STALL_FRONTEND_TLB	Frontend stall cycles, TLB	Counts when the frontend is stalled on any TLB misses being handled. This event also counts the...
0x8160	STALL_FRONTEND_CPUBOUND	Frontend stall cycles, processor bound	Counts cycles when the frontend could not send any micro-operations to the dispatch stage due to...
0x8161	STALL_FRONTEND_FLOW	Frontend stall cycles, flow control	Counts cycles when the frontend could not send any micro-operations to the dispatch stage due to...
0x8162	STALL_FRONTEND_FLUSH	Frontend stall cycles, flush recovery	Counts cycles when the frontend could not send any micro-operations to the dispatch stage as the...
0x8164	STALL_BACKEND_MEMBOUND	Backend stall cycles, memory bound	Counts cycles when the backend could not accept any micro-operations primarily due to memory...
0x8165	STALL_BACKEND_L1D	Backend stall cycles, level 1 data cache	Counts cycles when the backend is stalled, and there is a demand data miss in the L1 data cache.
0x8167	STALL_BACKEND_TLB	Backend stall cycles, TLB	Counts cycles when the backend is stalled on any demand TLB misses being handled.
0x8168	STALL_BACKEND_ST	Backend stall cycles, store	Counts cycles when the backend is stalled, and a store cannot be sent to the store buffer.
0x816A	STALL_BACKEND_CPUBOUND	Backend stall cycles, processor bound	Counts cycles when the backend could not accept any micro-operations due to any resource...
0x816B	STALL_BACKEND_BUSY	Backend stall cycles, backend busy	Counts cycles when the backend could not accept any micro-operations because of a lack of...

Code	Mnemonic	Name	Description
0x816C	STALL_BACKEND_ILOCK	Backend stall cycles, input dependency	Counts cycles when the backend could not accept any micro-operations because of an instruction...
0x816E	STALL_BACKEND_ATOMIC	Backend stall cycles, atomic operation	Counts cycles when the backend is stalled and an atomic operation is in progress.
0x816F	STALL_BACKEND_MEMCPYSET	Backend stall cycles, Memory Copy or Set operation	The counter counts each cycle counted by STALL_BACKEND_MEMBOUND when the backend is processing an...

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

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 dispatch logic 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
- frontend_core_bound
- frontend_mem_bound

Metric groups

Cycle_Accounting
Topdown_Frontend

Functional groups

Stall

0x0024 STALL_BACKEND, No operation sent for execution due to the backend, event

Counts cycles whenever the dispatch logic is unable to send any micro-operations to the backend of the pipeline because of backend resource constraints. Backend resource constraints can include issue stage fullness, execution stage fullness, or other internal pipeline resource fullness. All the backend slots were empty during the cycle when this event counts.

Related telemetry artifacts

Metrics

- backend_stalled_cycles
- backend_core_bound
- backend_mem_bound
- backend_stall_interlock_bound
- backend_stall_interlock_ls_bound
- backend_stall_interlock_ptr_chase_bound

- backend_stall_interlock_vpu_bound
- backend_busy_bound
- backend_busy_ls_bound
- backend_busy_vpu_arb_bound

Metric groups

Cycle_Accounting
Topdown_Backend

Functional groups

Stall

0x003C STALL, No operation sent for execution, event

Counts cycles when no operations are sent to the dispatch unit from the frontend or from the dispatch unit to the backend for any reason (either frontend or backend stall).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Stall

0x003D STALL_SLOT_BACKEND, No operation sent for execution on a Slot due to the backend, event

Counts slots per cycle in which an operation is available to the dispatch unit, but the operation is not sent from the dispatch unit to the backend due to backend resource constraints.

Related telemetry artifacts

Metrics

- backend_bound

Metric groups

Topdown_L1

Functional groups

Stall

0x003E STALL_SLOT_FRONTEND, No operation sent for execution on a Slot due to the frontend, event

Counts slots per cycle in which no operations are sent to the dispatch logic from the frontend due to frontend resource constraints.

Related telemetry artifacts

Metrics

- frontend_bound

Metric groups

Topdown_L1

Functional groups

Stall

0x003F STALL_SLOT, No operation sent for execution on a Slot, event

Counts slots per cycle in which no operations are sent to the dispatch logic from the frontend or from the dispatch logic to the backend for any reason (either frontend or backend stall).

Related telemetry artifacts

Metrics

- retiring
- bad_speculation

Metric groups

Topdown_L1

Functional groups

Stall

0x00EE IMP_STALL_SLOT_BACKEND_ILOCK, Backend slot stall, input dependency, event

Counts slots per cycle in which an operation is available to the dispatch unit, and the operation could not be sent to the backend due to an interlock between it and other operations in other slots in the dispatch unit.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Stall

0x00e6 IMP_STALL_BACKEND_ILOCK_VPU, Backend stall, input dependency from VPU instruction, event

Counts cycles when the backend is stalled because of an instruction interlock, and the source of at least one interlock is a vector or floating point instruction.

Related telemetry artifacts

Metrics

- backend_stall_interlock_vpu_bound

Metric groups

Topdown_Backend

Functional groups

Stall

`0x00ef` IMP_STALL_BACKEND_BUSY_VPU_ARB, Backend stall, busy VPU arbitration, event

Counts cycles when the backend could not accept any micro-operations because of a lack of arbitration to the shared vector/FP unit (VPU).

Related telemetry artifacts

Metrics

- backend_busy_vpu_arb_bound

Metric groups

Topdown_Backend

Functional groups

Stall

`0x00f1` IMP_STALL_BACKEND_BUSY_LS, Backend stall, busy LS unit, event

Counts cycles when the backend could not accept any micro-operations because of a lack of load-store (LS) execution resources. Refer to STALL_BACKEND_MEMBOUND to check whether the lack of resources is likely affected by memory resources such as caches or TLB. Refer to UNALIGNED_LDST_SPEC/LDST_SPEC to check whether the lack of resources is linked to load or store alignment.

Related telemetry artifacts

Metrics

- backend_busy_ls_bound

Metric groups

Topdown_Backend

Functional groups

Stall

`0x00f2` IMP_STALL_BACKEND_ILOCK_LS, Backend stall, input dependency from memory instruction, event

Counts cycles when the backend is stalled because of an instruction interlock, and the source of at least one interlock is a memory instruction. Refer to STALL_BACKEND_MEMBOUND to check whether the interlock is likely affected by memory resources such as caches or TLBs, or is instead primarily affected by load-to-use latency.

Related telemetry artifacts

Metrics

- backend_stall_interlock_ls_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x00f3 IMP_STALL_BACKEND_ILOCK_LS_INTO_AGU, Backend stall, memory instruction input dependency from memory instruction (pointer chase), event

Counts cycles when the backend is stalled because of an instruction interlock, the source of at least one interlock is a memory instruction, and the destination is the address of another memory instruction. Refer to STALL_BACKEND_MEMBOUND to check whether the interlock is likely affected by primarily memory resources such as caches or TLBs, or is instead primarily affected by load-to-use latency.

Related telemetry artifacts

Metrics

- backend_stall_interlock_ptr_chase_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x00f4 IMP_STALL_SLOT_BACKEND_PORT_CONTENTION, Backend slot stall, port contention, event

Counts slots per cycle in which an operation is available to the dispatch unit, and the operation could not be sent to the backend due to port contention between multiple operations in the dispatch unit.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Stall

0x3200 STALL_BACKEND_BUSY_CME, Backend stall cycles, SME2 unit busy, event

The counter counts each PE cycle counted by STALL_BACKEND_CPUBOUND when the PEs backend was stalled due SME instructions not able to progress, typically:

- When waiting for SME2 unit arbitration
- Because of SME2 unit backpressure
- Because instructions cannot be sent to the SME2 unit due to dependencies, commit, etc.

Related telemetry artifacts

Metrics

- backend_core_cme_bound
- backend_cme_cpu_bound
- backend_cme_backpressure_bound
- backend_cme_busy_arb_bound

Metric groups

Topdown_Backend

Topdown_CME

Functional groups

Stall

0x3201 STALL_BACKEND_BUSY_CMEBOUND, Backend stall cycles, SME2 unit backpressure, event

The counter counts each CPU cycle counted by STALL_BACKEND_BUSY_CME when the SME2 unit causes backpressure and does not accept instructions.

Notes:

- This event counts independently of oldest instruction being ready to be sent to the SME2 unit
- This event does not count when the CPU is waiting for arbitration, and STALL_BACKEND_BUSY_CME_ARB is counting.

Related telemetry artifacts

Metrics

- backend_cme_backpressure_bound

Metric groups

Topdown_CME

Functional groups

Stall

0x3202 STALL_BACKEND_BUSY_CME_ARB, Backend stall cycles, SME2 unit arbitration, event

The counter counts each CPU cycle counted by STALL_BACKEND_BUSY_CME when oldest SME instruction cannot be sent because it is waiting for arbitration.

Related telemetry artifacts

Metrics

- backend_cme_busy_arb_bound

Metric groups

Topdown_CME

Functional groups

Stall

0x3203 STALL_BACKEND_BUSY_CME_CPUBOUND, Backend stall cycles, SME2 unit stalled by CPU, event

The counter counts each PE cycle counted by STALL_BACKEND_BUSY_CME when not counted by STALL_BACKEND_BUSY_CME_BOUND or STALL_BACKEND_BUSY_CME_ARB.

This can typically be due to:

- Instruction waiting for commit point being reached
- A register dependency prevents the SSVE oldest instruction to be sent
- A control packet needs to be sent

Related telemetry artifacts

Metrics

- backend_cme_cpu_bound

Metric groups

Topdown_CME

Functional groups

Stall

`0x3204` STALL_BACKEND_ILOCK_FROM_CME, Backend stall, input dependency from SME2 unit, event

The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK when a CPU instruction waits for a register update from the SME2 unit, including predicate, flags and GPR updates.

Related telemetry artifacts

Metrics

- backend_stall_interlock_from_cme_bound
- backend_stall_interlock_from_cme_flags_bound
- backend_stall_interlock_from_cme_gpr_bound
- backend_stall_interlock_from_cme_predicate_bound

Metric groups

CME_Ilock_From_CME

Topdown_Backend

Functional groups

Stall

`0x3205` STALL_BACKEND_ILOCK_FROM_CME_GPR, Backend stall, input dependency from SME2 unit general purpose register, event

The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_FROM_CME when dependency is on a GPR register.

Related telemetry artifacts

Metrics

- backend_stall_interlock_from_cme_gpr_bound

Metric groups

CME_Ilock_From_CME

Functional groups

Stall

0x3206 STALL_BACKEND_ILOCK_FROM_CME_PRED, Backend stall, input dependency from SME2 unit predicate register, event

The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_FROM_CME when dependency is on a Predicate register.

Related telemetry artifacts

Metrics

- backend_stall_interlock_from_cme_predicate_bound

Metric groups

CME_Ilock_From_CME

Functional groups

Stall

0x3207 STALL_BACKEND_ILOCK_FROM_CME_FLAGS, Backend stall, input dependency from SME2 unit flag register, event

The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_FROM_CME when dependency is on flags register.

Related telemetry artifacts

Metrics

- backend_stall_interlock_from_cme_flags_bound

Metric groups

CME_Ilock_From_CME

Functional groups

Stall

0x3208 STALL_BACKEND_ILOCK_TO_CME, Backend stall, SME2 unit input dependency from CPU, event

The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK when oldest SME instruction is busy due to a data hazard between the SME2 unit and the CPU, including predicate updates and GPR updates.

Related telemetry artifacts

Metrics

- backend_stall_interlock_to_cme_bound
- backend_stall_interlock_to_cme_flags_bound
- backend_stall_interlock_to_cme_gpr_bound
- backend_stall_interlock_to_cme_predicate_bound

Metric groups

CME_Ilock_To_CME

Topdown_Backend

Functional groups

Stall

0x3209 STALL_BACKEND_ILOCK_TO_CME_GPR, Backend stall, SME2 unit input dependency from CPU on general purpose register, event

The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_TO_CME when oldest SME instruction is busy due to a GPR dependency.

Related telemetry artifacts

Metrics

- backend_stall_interlock_to_cme_gpr_bound

Metric groups

CME_Ilock_To_CME

Functional groups

Stall

0x320a STALL_BACKEND_ILOCK_TO_CME_PRED, Backend stall, SME2 unit input dependency from CPU predicate register, event

The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_TO_CME when operations when oldest SME instruction is busy due to a predicate register dependency.

Related telemetry artifacts

Metrics

- backend_stall_interlock_to_cme_predicate_bound

Metric groups

CME_Ilock_To_CME

Functional groups

Stall

0x320b STALL_BACKEND_ILOCK_TO_CME_FLAGS, Backend stall, SME2 unit input dependency from CPU on flags, event

The counter counts each CPU cycle counted by STALL_BACKEND_ILOCK_TO_CME when operations when oldest SME instruction is busy due to flags dependency.

Related telemetry artifacts

Metrics

- backend_stall_interlock_to_cme_flags_bound

Metric groups

CME_Ilock_To_CME

Functional groups

Stall

0x320c STALL_BACKEND_MEM_CME_LSRT_FULL, Backend stall cycles, SME2 unit stalled on LSRT full, event

The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when at least one Streaming SVE instruction is waiting for an entry being freed to be allocated into the LSRT.

Related telemetry artifacts

Metrics

- backend_mem_cme_lsrt_full_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x320d STALL_BACKEND_MEM_CME_BARRIER, Backend stall cycles, barrier stalled by SME2 unit, event

The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when a barrier instruction is executed and waits for completions from SME load/store transactions.

This includes effect due to store-release or load-acquire semantic.

Related telemetry artifacts

Metrics

- backend_mem_cme_barrier_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x320e STALL_BACKEND_MEM_CME_HZ_ON_CPU, Backend stall cycles, SME2 unit stalled by CPU memory hazard, event

The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when at least one Streaming SVE load/store instruction is waiting for resolution from an address hazard. This could be used to detect cases for which the CPU and SME unit are making overlapping accesses, that is, both are accessing the same location, and the SME unit cannot accept the operation to preserve the ordering of memory effects for the location required by the architecture.

Related telemetry artifacts

Metrics

- backend_mem_cme_hazard_cpu_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x320f STALL_BACKEND_MEM_CPU_HZ_ON_CME, Backend stall cycles, CPU stalled by SME2 unit memory hazard, event

The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when at least one CPU load/store instruction is waiting for resolution from an address hazard. This could be used to detect cases for which the CPU and SME unit are making overlapping accesses, that is, both are accessing the same location, and the CPU cannot execute the operation to preserve the ordering of memory effects for the location required by the architecture.

Related telemetry artifacts

Metrics

- backend_mem_cpu_hazard_cme_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x3210 STALL_BACKEND_MEM_CME, Backend stall cycles, SME2 unit stalled on memory hazard or LSRT full, event

The counter counts each CPU cycle counted by STALL_BACKEND_MEMBOUND when at least one Streaming SVE instruction is waiting for an entry being freed to be allocated into the LSRT or an address hazard to be resolved.

Related telemetry artifacts

Metrics

- backend_mem_cme_bound
- backend_mem_cme_hazard_cpu_bound
- backend_mem_cpu_hazard_cme_bound
- backend_mem_cme_lsrt_full_bound
- backend_mem_cme_barrier_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x4005 STALL_BACKEND_MEM, Memory stall cycles, event

Counts cycles when the backend is stalled, and there is a demand data miss in the last level core cache.

Related telemetry artifacts

Metrics

- backend_mem_cache_bound
- backend_cache_l1d_bound
- backend_cache_l2d_bound

Metric groups

Topdown_Backend

Functional groups

Stall

`0x8158` STALL_FRONTEND_MEMBOUND, Frontend stall cycles, memory bound, event

Counts cycles when the frontend could not send any micro-operations to the dispatch logic stage due to resource constraints in the memory resources.

Related telemetry artifacts

Metrics

- frontend_mem_bound
- frontend_mem_cache_bound
- frontend_mem_tlb_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

`0x8159` STALL_FRONTEND_L1I, Frontend stall cycles, level 1 instruction cache, event

Counts cycles when the frontend is stalled because there is an instruction fetch request pending in the L1 instruction cache.

Related telemetry artifacts

Metrics

- frontend_mem_cache_bound
- frontend_cache_l1i_bound
- frontend_cache_l2i_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

0x815B STALL_FRONTEND_MEM, Frontend stall cycles, last level PE cache or memory, event

Counts cycles when the frontend is stalled because there is an instruction fetch request pending in the last level core cache.

Related telemetry artifacts

Metrics

- frontend_mem_cache_bound
- frontend_cache_l1i_bound
- frontend_cache_l2i_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

0x815C STALL_FRONTEND_TLB, Frontend stall cycles, TLB, event

Counts when the frontend is stalled on any TLB misses being handled. This event also counts the TLB accesses made by hardware prefetches.

Related telemetry artifacts

Metrics

- frontend_mem_tlb_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

0x8160 STALL_FRONTEND_CPUBOUND, Frontend stall cycles, processor bound, event

Counts cycles when the frontend could not send any micro-operations to the dispatch stage due to resource constraints in the CPU resources excluding memory resources.

Related telemetry artifacts

Metrics

- frontend_core_bound
- frontend_core_flush_bound
- frontend_core_flow_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

`0x8161` STALL_FRONTEND_FLOW, Frontend stall cycles, flow control, event

Counts cycles when the frontend could not send any micro-operations to the dispatch stage due to resource constraints in the branch prediction unit.

Related telemetry artifacts

Metrics

- frontend_core_flow_bound

Metric groups

Topdown_Frontend

Functional groups

Stall

`0x8162` STALL_FRONTEND_FLUSH, Frontend stall cycles, flush recovery, event

Counts cycles when the frontend could not send any micro-operations to the dispatch stage as the frontend is recovering from a machine flush or resteer. Example scenarios that cause a flush include branch mispredictions, taken exceptions, micro-architectural flush etc.

Related telemetry artifacts

Metrics

- frontend_bound
- bad_speculation
- frontend_core_flush_bound

Metric groups

Topdown_Frontend
Topdown_L1

Functional groups

Stall

`0x8164` STALL_BACKEND_MEMBOUND, Backend stall cycles, memory bound, event

Counts cycles when the backend could not accept any micro-operations primarily due to memory pressure.

Related telemetry artifacts

Metrics

- backend_mem_bound
- backend_mem_cache_bound
- backend_mem_tlb_bound
- backend_mem_store_bound
- backend_mem_cme_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x8165 STALL_BACKEND_L1D, Backend stall cycles, level 1 data cache, event

Counts cycles when the backend is stalled, and there is a demand data miss in the L1 data cache.

Related telemetry artifacts

Metrics

- backend_mem_cache_bound
- backend_cache_l1d_bound
- backend_cache_l2d_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x8167 STALL_BACKEND_TLB, Backend stall cycles, TLB, event

Counts cycles when the backend is stalled on any demand TLB misses being handled.

Related telemetry artifacts

Metrics

- backend_mem_tlb_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x8168 STALL_BACKEND_ST, Backend stall cycles, store, event

Counts cycles when the backend is stalled, and a store cannot be sent to the store buffer.

Related telemetry artifacts

Metrics

- backend_mem_store_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x816A STALL_BACKEND_CPUBOUND, Backend stall cycles, processor bound, event

Counts cycles when the backend could not accept any micro-operations due to any resource constraints in the CPU.

Related telemetry artifacts

Metrics

- backend_core_bound
- backend_core_cme_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x816B STALL_BACKEND_BUSY, Backend stall cycles, backend busy, event

Counts cycles when the backend could not accept any micro-operations because of a lack of specific structural resources required for execution.

Related telemetry artifacts

Metrics

- backend_busy_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x816C STALL_BACKEND_ILOCK, Backend stall cycles, input dependency, event

Counts cycles when the backend could not accept any micro-operations because of an instruction interlock - that is, a dependency between instructions, such as a register dependency.

Related telemetry artifacts

Metrics

- backend_stall_interlock_bound
- backend_stall_interlock_from_cme_bound
- backend_stall_interlock_to_cme_bound

Metric groups

Topdown_Backend

Functional groups

Stall

0x816E STALL_BACKEND_ATOMIC, Backend stall cycles, atomic operation, event

Counts cycles when the backend is stalled and an atomic operation is in progress.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Stall

0x816F STALL_BACKEND_MEMCPYSET, Backend stall cycles, Memory Copy or Set operation, event

The counter counts each cycle counted by STALL_BACKEND_MEMBOUND when the backend is processing an Memory Copy or Set instruction. The Memory Copy instructions are CPY and CPYF. The Memory Set instructions are SET and SETG.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Stall

6.13 General (GENERAL) events for C1-Nano

General CPU related events.

Summary of events in General:

- Total implemented Common events: 3
- Total Implemented Product ImpDef events: 6
- PMU Only events : 6
- ETE Only events : 0

Table 6-13: 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...
0x3212	SM_ACTIVE_CYCLES	Cycles PSTATE.SM enabled	The counter counts each cycle counted by CPU_CYCLES when PSTATE.SM was enabled.
0x3213	CYCLES_CME_ALLOC	Cycles SME2 unit allocated	The counter counts each PE cycle counted by CPU_CYCLES where the CPU had an granted arbitration...
0x3214	CYCLES_ARB_PENDING_CME	Cycles SME2 unit arbitration pending	The counter counts each PE cycle counted by CPU_CYCLES where the CPU is in waiting for...
0x3215	CYCLES_CME_RECONNECT_PENDING	Cycles SME2 unit reconnect pending	The counter counts each PE cycle counted by CYCLES_ARB_PENDING_CME where the CPU is in waiting...
0x3216	ARB_CME_COUNT	SME2 unit arbitration requests	The counter counts the number of times an arbitration to SME2 unit is requested, either an...
0x3217	RECONNECT_CME_COUNT	SME2 unit reconnect requests	The counter counts the number of times the CPU needs to request arbitration following...
0x4004	CNT_CYCLES	Constant frequency cycles	Counts constant frequency cycles

Code	Mnemonic	Name	Description
0x8380	ZA_ACTIVE	PSTATE.ZA active cycles	The counter counts each cycle counted by CPU_CYCLES when PSTATE.ZA was enabled.

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x0011 CPU_CYCLES, Cycle, event

Counts CPU clock cycles (not timer cycles). The clock measured by this event is defined as the physical clock driving the CPU logic.

Related telemetry artifacts

Metrics

- frontend_stalled_cycles
- backend_stalled_cycles
- frontend_bound
- backend_bound
- retiring
- bad_speculation
- ipc
- sm_active_cycles_ratio
- za_active_cycles_ratio
- cme_alloc_cycles_ratio
- cme_arb_pending_ratio

Metric groups

Cycle_Accounting
General
Topdown_L1

Functional groups

General

0x3212 SM_ACTIVE_CYCLES, Cycles PSTATE.SM enabled, event

The counter counts each cycle counted by CPU_CYCLES when PSTATE.SM was enabled.

Related telemetry artifacts

Metrics

- sm_active_cycles_ratio

Metric groups

Cycle_Accounting

Functional groups

General

0x3213 CYCLES_CME_ALLOC, Cycles SME2 unit allocated, event

The counter counts each PE cycle counted by CPU_CYCLES where the CPU had an granted arbitration to the SME2 unit, such that the SME2 and Streaming SVE state of the CPU is held in that SME2 unit. It does not count cycles during which a CPU has requested arbitration and is waiting for acknowledgement.

Related telemetry artifacts

Metrics

- cme_alloc_cycles_ratio

Metric groups

Cycle_Accounting

Functional groups

General

0x3214 CYCLES_ARB_PENDING_CME, Cycles SME2 unit arbitration pending, event

The counter counts each PE cycle counted by CPU_CYCLES where the CPU is in waiting for arbitration while attempting to access the SME2 unit. If can be due an initial arbitration request or contention.

Related telemetry artifacts

Metrics

- cme_arb_pending_ratio

Metric groups

Cycle_Accounting

Functional groups

General

0x3215 CYCLES_CME_RECONNECT_PENDING, Cycles SME2 unit reconnect pending, event

The counter counts each PE cycle counted by CYCLES_ARB_PENDING_CME where the CPU is in waiting for arbitration due to contention, when it was previously arbitrated to a SME2 unit but arbitration was granted to another CPU.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

General

0x3216 ARB_CME_COUNT, SME2 unit arbitration requests, event

The counter counts the number of times an arbitration to SME2 unit is requested, either an initial request, or a reconnect request due to contention.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

General

0x3217 RECONNECT_CME_COUNT, SME2 unit reconnect requests, event

The counter counts the number of times the CPU needs to request arbitration following a disconnect due to contention.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

General

0x4004 CNT_CYCLES, Constant frequency cycles, event

Counts constant frequency cycles

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

General

0x8380 ZA_ACTIVE, PSTATE.ZA active cycles, event

The counter counts each cycle counted by CPU_CYCLES when PSTATE.ZA was enabled.

Related telemetry artifacts

Metrics

- za_active_cycles_ratio

Metric groups

Cycle_Accounting

Functional groups

General

6.14 TLB (TLB) events for C1-Nano

TLB and MMU related events.

Summary of events in TLB:

- Total implemented Common events: 29
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-14: TLB events summary

Code	Mnemonic	Name	Description
0x0002	L1I_TLB_REFILL	Level 1 instruction TLB refill	Counts L1 instruction TLB refills from any Instruction fetch. If there are multiple misses in the...
0x0005	L1D_TLB_REFILL	Level 1 data TLB refill	Counts L1 data TLB accesses that resulted in TLB refills. If there are multiple misses in the TLB...
0x0025	L1D_TLB	Level 1 data TLB access	Counts L1 data TLB accesses caused by any memory load or store operation. Note that load or store...
0x0026	L1I_TLB	Level 1 instruction TLB access	Counts L1 instruction TLB accesses, whether the access hits or misses in the TLB. This event...
0x002D	L2D_TLB_REFILL	Level 2 data TLB refill	Counts L2 TLB refills caused by memory operations from both data and instruction fetch, except...
0x002E	L2I_TLB_REFILL	Level 2 instruction TLB refill	Counts L2 TLB refills caused by instruction memory accesses except for those caused by TLB...
0x002F	L2D_TLB	Level 2 data TLB access	Counts L2 TLB accesses except those caused by TLB maintenance operations.
0x0030	L2I_TLB	Level 2 instruction TLB access	Counts L2 TLB accesses caused by instruction memory access except for those caused by TLB...
0x0034	DTLB_WALK	Data TLB access with at least one translation table walk	Counts data memory translation table walks caused by a miss in the L2 TLB driven by a memory...
0x0035	ITLB_WALK	Instruction TLB access with at least one translation table walk	Counts instruction memory translation table walks caused by a miss in the L2 TLB driven by a...
0x8128	DTLB_WALK_PERCYC	Event in progress, DTLB WALK	Counts number of DTLB translation table walks in progress in a cycle.
0x8129	ITLB_WALK_PERCYC	Event in progress, ITLB WALK	Counts number of ITLB translation table walks in progress in a cycle.
0x8131	L1I_TLB_RD	Level 1 instruction TLB demand access	Counts L1 instruction TLB demand accesses whether the access hits or misses in the TLB.
0x8134	DTLB_HWUPD	Data TLB hardware update of translation table	Counts cycles during which there is a data translation table access that caused an update of the...
0x8135	ITLB_HWUPD	Instruction TLB hardware update of translation table	Counts cycles during which there was an instruction translation table access that caused an...
0x8136	DTLB_STEP	Data TLB translation table walk, step	Counts number of page table walks taken for a data translation table miss that caused a refill in...
0x8137	ITLB_STEP	Instruction TLB translation table walk, step	Counts number of page table walks taken for an instruction translation table miss that caused a...

Code	Mnemonic	Name	Description
0x8138	DTLB_WALK_LARGE	Data TLB large page translation table walk	Counts data memory accesses caused page translation table walks that yield a large page. Partial...
0x8139	ITLB_WALK_LARGE	Instruction TLB large page translation table walk	Counts instruction memory access caused page translation table walks that yields a large page....
0x813A	DTLB_WALK_SMALL	Data TLB small page translation table walk	Counts data memory access caused page translation table walks that yield a small page. Partial...
0x813B	ITLB_WALK_SMALL	Instruction TLB small page translation table walk	Counts instruction memory access caused translation table walks that yield a small page. Partial...
0x813C	DTLB_WALK_RW	Data TLB demand access with at least one translation table walk	Counts demand data TLB accesses that resulted in at least one translation table walk. Partial...
0x813D	ITLB_WALK_RD	Instruction TLB demand access with at least one translation table walk	Counts demand instruction TLB accesses that resulted in at least one translation table walk....
0x8188	DTLB_WALK_BLOCK	Data TLB block translation table walk	Counts number of demand data translation table walks caused by a miss in the L2 TLB and yielding...
0x8189	ITLB_WALK_BLOCK	Instruction TLB block translation table walk	Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding...
0x818A	DTLB_WALK_PAGE	Data TLB page translation table walk	Counts number of demand data translation table walks caused by a miss in the L2 TLB and yielding...
0x818B	ITLB_WALK_PAGE	Instruction TLB page translation table walk	Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding...
0x82FA	DTLB_WALK_HWPRF	Data TLB hardware prefetch access with at least one translation table walk	This event counts any access counted by DTLB_WALK that is due to a hardware prefetch
0x82FE	L1D_TLB_HWPRF	Level 1 data TLB access, hardware prefetch	This event counts L1 data TLB accesses due to a hardware prefetch.

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x0002 L1I_TLB_REFILL, Level 1 instruction TLB refill, event

Counts L1 instruction TLB refills from any Instruction fetch. If there are multiple misses in the TLB that are resolved by the refill, then this event only counts once. This event will not count if the translation table walk results in a fault (such as a translation or access fault), since there is no new translation created for the TLB.

Related telemetry artifacts

Metrics

- l1i_tlb_mpki in ITLB_Effectiveness
- l1i_tlb_mpki in MPKI
- l1i_tlb_miss_ratio in ITLB_Effectiveness
- l1i_tlb_miss_ratio in Miss_Ratio

Metric groups

ITLB_Effectiveness
MPKI
Miss_Ratio

Functional groups

TLB

0x0005 L1D_TLB_REFILL, Level 1 data TLB refill, event

Counts L1 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 L1 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
- dtlb_walk_page_ratio
- dtlb_walk_block_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 L1 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
- itlb_walk_page_ratio
- itlb_walk_block_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 L2 TLB refills caused by memory operations from both data and instruction fetch, except for those caused by TLB maintenance operations and hardware prefetches.

Related telemetry artifacts

Metrics

- l2_tlb_mpki in DTLB_Effectiveness
- l2_tlb_mpki in ITLB_Effectiveness
- l2_tlb_mpki in MPKI
- l2_tlb_miss_ratio in DTLB_Effectiveness
- l2_tlb_miss_ratio in ITLB_Effectiveness
- l2_tlb_miss_ratio in Miss_Ratio

Metric groups

DTLB_Effectiveness
ITLB_Effectiveness
MPKI
Miss_Ratio

Functional groups

TLB

0x002E L2I_TLB_REFILL, Level 2 instruction TLB refill, event

Counts L2 TLB refills caused by instruction memory accesses 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 this Telemetry Specification.

Functional groups

TLB

0x002F L2D_TLB, Level 2 data TLB access, event

Counts L2 TLB accesses except those caused by TLB maintenance operations.

Related telemetry artifacts

Metrics

- l2_tlb_miss_ratio in DTLB_Effectiveness
- l2_tlb_miss_ratio in ITLB_Effectiveness
- l2_tlb_miss_ratio in Miss_Ratio

Metric groups

DTLB_Effectiveness
ITLB_Effectiveness
Miss_Ratio

Functional groups

TLB

0x0030 L2I_TLB, Level 2 instruction TLB access, event

Counts L2 TLB accesses caused by instruction memory access 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 this Telemetry Specification.

Functional groups

TLB

0x0034 DTLB_WALK, Data TLB access with at least one translation table walk, event

Counts data memory translation table walks caused by a miss in the L2 TLB driven by a memory access. Note that partial translations that also cause a table walk are counted. This event does not count table walks caused 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
- dtlb_walk_average_depth
- dtlb_walk_average_latency in Average_Latency
- dtlb_walk_average_latency in DTLB_Effectiveness

Metric groups

Average_Latency
DTLB_Effectiveness
MPKI
Miss_Ratio

Functional groups

TLB

`0x0035` ITLB_WALK, Instruction TLB access with at least one translation table walk, event

Counts instruction memory translation table walks caused by a miss in the L2 TLB driven by a memory access. Partial translations that also cause a table walk are counted. This event does not count table walks caused 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
- itlb_walk_average_depth
- itlb_walk_average_latency in Average_Latency
- itlb_walk_average_latency in ITLB_Effectiveness

Metric groups

Average_Latency
ITLB_Effectiveness
MPKI
Miss_Ratio

Functional groups

TLB

`0x8128` DTLB_WALK_PERCYC, Event in progress, DTLB WALK, event

Counts number of DTLB translation table walks in progress in a cycle.

Related telemetry artifacts

Metrics

- dtlb_walk_average_latency in Average_Latency
- dtlb_walk_average_latency in DTLB_Effectiveness

Metric groups

Average_Latency
DTLB_Effectiveness

Functional groups

TLB

`0x8129` ITLB_WALK_PERCYC, Event in progress, ITLB WALK, event

Counts number of ITLB translation table walks in progress in a cycle.

Related telemetry artifacts

Metrics

- itlb_walk_average_latency in Average_Latency
- itlb_walk_average_latency in ITLB_Effectiveness

Metric groups

Average_Latency
ITLB_Effectiveness

Functional groups

TLB

`0x8131` L1I_TLB_RD, Level 1 instruction TLB demand access, event

Counts L1 instruction TLB demand accesses whether the access hits or misses in the TLB.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

`0x8134` DTLB_HWUPD, Data TLB hardware update of translation table, event

Counts cycles during which there is a data translation table access that caused an update of the translation table.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

0x8135 ITLB_HWUPD, Instruction TLB hardware update of translation table, event

Counts cycles during which there was an instruction translation table access that caused an update of the translation table.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

0x8136 DTLB_STEP, Data TLB translation table walk, step, event

Counts number of page table walks taken for a data translation table miss that caused a refill in the last level TLB.

Related telemetry artifacts

Metrics

- dtlb_walk_average_depth

Metric groups

DTLB_Effectiveness

Functional groups

TLB

0x8137 ITLB_STEP, Instruction TLB translation table walk, step, event

Counts number of page table walks taken for an instruction translation table miss that caused a refill in the last level TLB.

Related telemetry artifacts

Metrics

- itlb_walk_average_depth

Metric groups

ITLB_Effectiveness

Functional groups

TLB

0x8138 DTLB_WALK_LARGE, Data TLB large page translation table walk, event

Counts data memory accesses caused page translation table walks that yield a large page. Partial translations that also cause a table walk are counted. This event does not count table walks caused by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

0x8139 ITLB_WALK_LARGE, Instruction TLB large page translation table walk, event

Counts instruction memory access caused page translation table walks that yields a large page. Partial translations that also cause a table walk are counted. This event does not count table walks caused by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

0x813A DTLB_WALK_SMALL, Data TLB small page translation table walk, event

Counts data memory access caused page translation table walks that yield a small page. Partial translations that also cause a table walk are counted. This event does not count table walks caused by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

0x813B ITLB_WALK_SMALL, Instruction TLB small page translation table walk, event

Counts instruction memory access caused translation table walks that yield a small page. Partial translations that also cause a table walk are counted. This event does not count table walks caused by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

0x813C DTLB_WALK_RW, Data TLB demand access with at least one translation table walk, event

Counts demand data TLB accesses that resulted in at least one translation table walk. Partial translations that also cause a table walk are counted. This event does not count table walks caused by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

0x813D ITLB_WALK_RD, Instruction TLB demand access with at least one translation table walk, event

Counts demand instruction TLB accesses that resulted in at least one translation table walk. Partial translations that also cause a table walk are counted. This event does not count table walks caused by TLB maintenance operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

0x8188 DTLB_WALK_BLOCK, Data TLB block translation table walk, event

Counts number of demand data translation table walks caused by a miss in the L2 TLB and yielding a block descriptor at level 1 or level 2 of the translation.

Related telemetry artifacts

Metrics

- dtlb_walk_block_ratio

Metric groups

DTLB_Effectiveness

Functional groups

TLB

0x8189 ITLB_WALK_BLOCK, Instruction TLB block translation table walk, event

Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding a block descriptor at level 1 or level 2 of the translation.

Related telemetry artifacts

Metrics

- itlb_walk_block_ratio

Metric groups

ITLB_Effectiveness

Functional groups

TLB

0x818A DTLB_WALK_PAGE, Data TLB page translation table walk, event

Counts number of demand data translation table walks caused by a miss in the L2 TLB and yielding a page descriptor at level 1 or level 2 of the translation.

Related telemetry artifacts

Metrics

- dtlb_walk_page_ratio

Metric groups

DTLB_Effectiveness

Functional groups

TLB

0x818B ITLB_WALK_PAGE, Instruction TLB page translation table walk, event

Counts number of instruction translation table walks caused by a miss in the L2 TLB and yielding a page descriptor at level 3 of the translation.

Related telemetry artifacts

Metrics

- itlb_walk_page_ratio

Metric groups

ITLB_Effectiveness

Functional groups

TLB

0x82FA DTLB_WALK_HWPRF, Data TLB hardware prefetch access with at least one translation table walk, event

This event counts any access counted by DTLB_WALK that is due to a hardware prefetch

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

0x82FE L1D_TLB_HWPRF, Level 1 data TLB access, hardware prefetch, event

This event counts L1 data TLB accesses due to a hardware prefetch.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TLB

6.15 SVE (SVE) events for C1-Nano

SVE related events.

Summary of events in SVE:

- Total implemented Common events: 16
- Total Implemented Product ImpDef events: 5

- PMU Only events : 5
- ETE Only events : 0

Table 6-15: SVE events summary

Code	Mnemonic	Name	Description
0x3234	SSVE_PRED_SPEC	Operation speculatively executed, Streaming SVE predicated	Counts operations counted by SVE_PRED_SPEC, but in Streaming mode only. Note: this counts SME...
0x3235	SSVE_PRED_EMPTY_SPEC	Operation speculatively executed, Streaming SVE predicated with no active predicates	Counts operations counted by SVE_PRED_EMPTY_SPEC, but in Streaming mode only.
0x3236	SSVE_PRED_FULL_SPEC	Operation speculatively executed, Streaming SVE predicated with all active predicates	Counts speculatively executed predicated SVE operations with all predicate elements active,...
0x3237	SSVE_PRED_NOT_FULL_SPEC	Streaming SVE predicated operations Speculatively executed with no active or partially active predicates	Counts speculatively executed predicated SVE operations with at least one non active predicate...
0x3238	SSVE_PRED_PARTIAL_SPEC	Operation speculatively executed, Streaming SVE predicated with partially active predicates	Counts speculatively executed predicated SVE operations with at least one but not all active...
0x8006	SVE_INST_SPEC	Operation speculatively executed, SVE, including load and store	The counter counts each Speculatively executed operation due to an SVE instruction. It is...
0x8014	FP_HP_SPEC	Floating-point operation speculatively executed, half precision	Counts speculatively executed half precision floating point operations.
0x8018	FP_SP_SPEC	Floating-point operation speculatively executed, single precision	Counts speculatively executed single precision floating point operations.
0x801C	FP_DP_SPEC	Floating-point operation speculatively executed, double precision	Counts speculatively executed double precision floating point operations.
0x8074	SVE_PRED_SPEC	Operation speculatively executed, SVE predicated	Counts speculatively executed predicated SVE operations.
0x8075	SVE_PRED_EMPTY_SPEC	Operation speculatively executed, SVE predicated with no active predicates	Counts speculatively executed predicated SVE operations with no active predicate elements.
0x8076	SVE_PRED_FULL_SPEC	Operation speculatively executed, SVE predicated with all active predicates	Counts speculatively executed predicated SVE operations with all predicate elements active.
0x8077	SVE_PRED_PARTIAL_SPEC	Operation speculatively executed, SVE predicated with partially active predicates	Counts speculatively executed predicated SVE operations with at least one but not all active...
0x8078	SVE_UNPRED_SPEC	Operation speculatively executed, SVE unpredicated	The counter counts each Speculatively executed SIMD data-processing, load, or store operation...
0x8079	SVE_PRED_NOT_FULL_SPEC	SVE predicated operations speculatively executed with no active or partially active predicates	Counts speculatively executed predicated SVE operations with at least one non active predicate...
0x80BC	SVE_LDFF_SPEC	Operation speculatively executed, SVE first-fault load	Counts speculatively executed SVE first fault or non-fault load operations.
0x80BD	SVE_LDFF_FAULT_SPEC	Operation speculatively executed, SVE first-fault load which set FFR bit to 0b0	Counts speculatively executed SVE first fault or non-fault load operations that clear at least...

Code	Mnemonic	Name	Description
0x80E3	ASE_SVE_INT8_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 8-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...
0x80E7	ASE_SVE_INT16_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 16-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...
0x80EB	ASE_SVE_INT32_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 32-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...
0x80EF	ASE_SVE_INT64_SPEC	Integer operation speculatively executed, Advanced SIMD or SVE 64-bit	Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type...

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x3234 SSVE_PRED_SPEC, Operation speculatively executed, Streaming SVE predicated, event

Counts operations counted by SVE_PRED_SPEC, but in Streaming mode only.

Note: this counts SME operations requiring PSTATE.ZA to be set, including 2D operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x3235 SSVE_PRED_EMPTY_SPEC, Operation speculatively executed, Streaming SVE predicated with no active predicates, event

Counts operations counted by SVE_PRED_EMPTY_SPEC, but in Streaming mode only.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x3236 SSVE_PRED_FULL_SPEC, Operation speculatively executed, Streaming SVE predicated with all active predicates, event

Counts speculatively executed predicated SVE operations with all predicate elements active, specifically in Streaming mode.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x3237 SSVE_PRED_NOT_FULL_SPEC, Streaming SVE predicated operations Speculatively executed with no active or partially active predicates, event

Counts speculatively executed predicated SVE operations with at least one non active predicate elements, specifically in Streaming mode.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x3238 SSVE_PRED_PARTIAL_SPEC, Operation speculatively executed, Streaming SVE predicated with partially active predicates, event

Counts speculatively executed predicated SVE operations with at least one but not all active predicate elements, specifically in streaming mode.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x8006 SVE_INST_SPEC, Operation speculatively executed, SVE, including load and store, event

The counter counts each Speculatively executed operation due to an SVE instruction.

It is **IMPLEMENTATION DEFINED** whether the counter counts operations due to non-SIMD SVE instructions.

Instructions classified as SME instructions and counted by SME_INST_SPEC are not counted by this event.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Spec_Operation

SVE

0x8014 FP_HP_SPEC, Floating-point operation speculatively executed, half precision, event

Counts speculatively executed half precision floating point operations.

Related telemetry artifacts

Metrics

- fp16_percentage

Metric groups

FP_Precision_Mix

Functional groups

SVE

`0x8018` FP_SP_SPEC, Floating-point operation speculatively executed, single precision, event

Counts speculatively executed single precision floating point operations.

Related telemetry artifacts

Metrics

- fp32_percentage

Metric groups

FP_Precision_Mix

Functional groups

SVE

`0x801C` FP_DP_SPEC, Floating-point operation speculatively executed, double precision, event

Counts speculatively executed double precision floating point operations.

Related telemetry artifacts

Metrics

- fp64_percentage

Metric groups

FP_Precision_Mix

Functional groups

SVE

`0x8074` SVE_PRED_SPEC, Operation speculatively executed, SVE predicated, event

Counts speculatively executed predicated SVE operations.

Related telemetry artifacts

Metrics

- sve_predicate_percentage
- sve_predicate_full_percentage
- sve_predicate_partial_percentage
- sve_predicate_empty_percentage

Metric groups

SVE_Effectiveness

Functional groups

SVE

0x8075 SVE_PRED_EMPTY_SPEC, Operation speculatively executed, SVE predicated with no active predicates, event

Counts speculatively executed predicated SVE operations with no active predicate elements.

Related telemetry artifacts

Metrics

- sve_predicate_empty_percentage

Metric groups

SVE_Effectiveness

Functional groups

SVE

0x8076 SVE_PRED_FULL_SPEC, Operation speculatively executed, SVE predicated with all active predicates, event

Counts speculatively executed predicated SVE operations with all predicate elements active.

Related telemetry artifacts

Metrics

- sve_predicate_full_percentage

Metric groups

SVE_Effectiveness

Functional groups

SVE

0x8077 SVE_PRED_PARTIAL_SPEC, Operation speculatively executed, SVE predicated with partially active predicates, event

Counts speculatively executed predicated SVE operations with at least one but not all active predicate elements.

Related telemetry artifacts

Metrics

- sve_predicate_partial_percentage

Metric groups

SVE_Effectiveness

Functional groups

SVE

0x8078 SVE_UNPRED_SPEC, Operation speculatively executed, SVE unpredicated, event

The counter counts each Speculatively executed SIMD data-processing, load, or store operation due to an SVE instruction without a Governing predicate operand.

This counts the SME operations requiring PSTATE.ZA to be set. It does not count Advanced SIMD operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x8079 SVE_PRED_NOT_FULL_SPEC, SVE predicated operations speculatively executed with no active or partially active predicates, event

Counts speculatively executed predicated SVE operations with at least one non active predicate elements.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x80BC SVE_LDFF_SPEC, Operation speculatively executed, SVE first-fault load, event

Counts speculatively executed SVE first fault or non-fault load operations.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x80BD SVE_LDFF_FAULT_SPEC, Operation speculatively executed, SVE first-fault load which set FFR bit to 0b0, event

Counts speculatively executed SVE first fault or non-fault load operations that clear at least one bit in the FFR.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x80E3 ASE_SVE_INT8_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 8-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type an 8-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x80E7 ASE_SVE_INT16_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 16-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type a 16-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x80EB ASE_SVE_INT32_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 32-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type a 32-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

0x80EF ASE_SVE_INT64_SPEC, Integer operation speculatively executed, Advanced SIMD or SVE 64-bit, event

Counts speculatively executed Advanced SIMD or SVE integer operations with the largest data type a 64-bit integer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

SVE

6.16 Non_PMU (NON_PMU) events for C1-Nano

Non-PMU related events.

Summary of events in Non_PMU:

- Total implemented Common events: 4
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 2

Table 6-16: Non_PMU events summary

Code	Mnemonic	Name	Description
0x400C	TRB_WRAP	Trace buffer current write pointer wrapped	This event is generated each time the current write pointer is wrapped to the base pointer.
0x400D	PMU_OVFS	PMU overflow, counters accessible to EL1 and EL0	This event is generated each time an event causes a PMEVCTNR<n>_EL1 counter overflow when...
0x400E	TRB_TRIG	Trace buffer Trigger Event	This event is generated when a Trace Buffer Extension Trigger Event occurs.
0x400F	PMU_HOVFS	PMU overflow, counters reserved for use by EL2	This event is generated each time an event causes a PMEVCTNR<n>_EL1 counter overflow when...

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x400C TRB_WRAP, Trace buffer current write pointer wrapped, event

This event is generated each time the current write pointer is wrapped to the base pointer.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Non_PMU

0x400D PMU_OVFS, PMU overflow, counters accessible to EL1 and EL0, event

This event is generated each time an event causes a PMEVCTNR<n>_EL1 counter overflow when PMINTENSET_EL1[n] is set to 1, for each implemented PMU counter n in the range 0 <= n < UInt(MDCR_EL2.HPMN), and the Cycle Counter (n = 31).

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Non_PMU

0x400E TRB_TRIG, Trace buffer Trigger Event, event

This event is generated when a Trace Buffer Extension Trigger Event occurs.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Non_PMU

0x400F PMU_HOVFS, PMU overflow, counters reserved for use by EL2, event

This event is generated each time an event causes a PMEVCTNR<n>_EL1 counter overflow when PMINTENSET_EL1[n] is set to 1, for each implemented PMU counter n in the range UInt(MDCR_EL2.HPMN) <= n < UInt(PMCR_EL0.N). This event is not transmitted to a PE Trace Unit while TRCFR_EL2.E2TRE == 0b0.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Non_PMU

6.17 TRCEXT (TRCEXT) events for C1-Nano

External trace related events.

Summary of events in TRCEXT:

- Total implemented Common events: 8
- Total Implemented Product ImpDef events: 0
- PMU Only events : 0
- ETE Only events : 0

Table 6-17: TRCEXT events summary

Code	Mnemonic	Name	Description
0x4010	TRCEXTOUT0	Trace unit external output 0	This event is generated each time an event is signaled by ETE external event 0.
0x4011	TRCEXTOUT1	Trace unit external output 1	This event is generated each time an event is signaled by ETE external event 1.
0x4012	TRCEXTOUT2	Trace unit external output 2	This event is generated each time an event is signaled by ETE external event 2.
0x4013	TRCEXTOUT3	Trace unit external output 3	This event is generated each time an event is signaled by ETE external event 3.
0x4018	CTI_TRIGOUT4	Cross-trigger Interface output trigger 4	This event is generated each time an event is signaled on CTI output trigger 4.

Code	Mnemonic	Name	Description
0x4019	CTI_TRIGOUT5	Cross-trigger Interface output trigger 5	This event is generated each time an event is signaled on CTI output trigger 5.
0x401A	CTI_TRIGOUT6	Cross-trigger Interface output trigger 6	This event is generated each time an event is signaled on CTI output trigger 6.
0x401B	CTI_TRIGOUT7	Cross-trigger Interface output trigger 7	This event is generated each time an event is signaled on CTI output trigger 7.

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x4010 TRCEXTOUT0, Trace unit external output 0, event

This event is generated each time an event is signaled by ETE external event 0.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TRCEXT

0x4011 TRCEXTOUT1, Trace unit external output 1, event

This event is generated each time an event is signaled by ETE external event 1.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TRCEXT

0x4012 TRCEXTOUT2, Trace unit external output 2, event

This event is generated each time an event is signaled by ETE external event 2.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TRCEXT

0x4013 TRCEXTOUT3, Trace unit external output 3, event

This event is generated each time an event is signaled by ETE external event 3.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TRCEXT

0x4018 CTI_TRIGOUT4, Cross-trigger Interface output trigger 4, event

This event is generated each time an event is signaled on CTI output trigger 4.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TRCEXT

0x4019 CTI_TRIGOUT5, Cross-trigger Interface output trigger 5, event

This event is generated each time an event is signaled on CTI output trigger 5.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TRCEXT

0x401A CTI_TRIGOUT6, Cross-trigger Interface output trigger 6, event

This event is generated each time an event is signaled on CTI output trigger 6.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TRCEXT

0x401B CTI_TRIGOUT7, Cross-trigger Interface output trigger 7, event

This event is generated each time an event is signaled on CTI output trigger 7.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

TRCEXT

6.18 Coherency (COHERENCY") events for C1-Nano

Coherency related events.

Summary of events in Coherency:

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

- ETE Only events : 0

Table 6-18: Coherency events summary

Code	Mnemonic	Name	Description
0x8190	ISNP_HIT_RD	Snoop hit, demand instruction fetch	The counter counts each snoop generated in response to a demand Instruction memory access that...
0x8191	ISNP_HIT_NEAR_RD	Snoop hit in near cache, demand instruction fetch	The counter counts each snoop generated in response to a demand Instruction memory access counted...
0x8192	ISNP_HIT_FAR_RD	Snoop hit in far cache, demand instruction fetch	The counter counts each snoop generated in response to a demand Instruction memory access counted...
0x8194	DSNP_HIT_RD	Snoop hit, demand data read	The counter counts each snoop generated in response to a demand Memory-read operation counted by...
0x8195	DSNP_HIT_NEAR_RD	Snoop hit in near cache, demand data read	The counter counts each snoop generated in response to a demand Memory-read operation counted by...
0x8196	DSNP_HIT_FAR_RD	Snoop hit in far cache, demand data read	The counter counts each snoop generated in response to a demand Memory-read operation counted by...
0x8198	DSNP_HIT_WR	Snoop hit, demand data write	The counter counts each snoop generated in response to a demand Memory-write operation counted by...
0x8199	DSNP_HIT_NEAR_WR	Snoop hit in near cache, demand data write	The counter counts each snoop generated in response to a demand Memory-write operation counted by...
0x819A	DSNP_HIT_FAR_WR	Snoop hit in far cache, demand data write	The counter counts each snoop generated in response to a demand Memory-write operation counted by...
0x819C	DSNP_HIT_RW	Snoop hit, demand data access	The counter counts each snoop generated in response to a demand Memory-read operation or demand...
0x819D	DSNP_HIT_NEAR_RW	Snoop hit in near cache, demand data access	The counter counts each snoop generated in response to a demand Memory-read operation or demand...
0x819E	DSNP_HIT_FAR_RW	Snoop hit in far cache, demand data access	The counter counts each snoop generated in response to a demand Memory-read operation or demand...
0x81B4	DSNP_HIT	Snoop hit, data	The counter counts each snoop that hits in a cache outside of the cache hierarchy of this PE.
0x81B5	DSNP_HIT_NEAR	Snoop hit in near cache, data	The counter counts each snoop counted by DSNP_HIT that hits in a cache outside of the cache...
0x81B6	DSNP_HIT_FAR	Snoop hit in far cache, data	The counter counts each snoop counted by DSNP_HIT that hits in a cache outside the local PE...

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x8190 ISNP_HIT_RD, Snoop hit, demand instruction fetch, event

The counter counts each snoop generated in response to a demand Instruction memory access that hits in a cache outside of the cache hierarchy of this PE.

Related telemetry artifacts

Metrics

- system_peer_cluster_cache_hit_ratio

Metric groups

System_Memory_Effectiveness

Functional groups

Coherency

`0x8191` ISNP_HIT_NEAR_RD, Snoop hit in near cache, demand instruction fetch, event

The counter counts each snoop generated in response to a demand Instruction memory access counted by ISNP_HIT_RD that hits in a cache outside of the cache hierarchy of this PE in the local PE cluster.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

`0x8192` ISNP_HIT_FAR_RD, Snoop hit in far cache, demand instruction fetch, event

The counter counts each snoop generated in response to a demand Instruction memory access counted by ISNP_HIT_RD that hits in a cache outside the local PE cluster on the same device.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

`0x8194` DSNP_HIT_RD, Snoop hit, demand data read, event

The counter counts each snoop generated in response to a demand Memory-read operation counted by DSNP_HIT_RW that hits in a cache outside of the cache hierarchy of this PE.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

`0x8195` DSNP_HIT_NEAR_RD, Snoop hit in near cache, demand data read, event

The counter counts each snoop generated in response to a demand Memory-read operation counted by DSNP_HIT_RD that hits in a cache outside of the cache hierarchy of this PE in the local PE cluster.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

0x8196 DSNP_HIT_FAR_RD, Snoop hit in far cache, demand data read, event

The counter counts each snoop generated in response to a demand Memory-read operation counted by DSNP_HIT_RD that hits in a cache outside the local PE cluster on the same device.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

0x8198 DSNP_HIT_WR, Snoop hit, demand data write, event

The counter counts each snoop generated in response to a demand Memory-write operation counted by DSNP_HIT_RW that hits in a cache outside of the cache hierarchy of this PE.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

0x8199 DSNP_HIT_NEAR_WR, Snoop hit in near cache, demand data write, event

The counter counts each snoop generated in response to a demand Memory-write operation counted by DSNP_HIT_WR that hits in a cache outside of the cache hierarchy of this PE in the local PE cluster.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

0x819A DSNP_HIT_FAR_WR, Snoop hit in far cache, demand data write, event

The counter counts each snoop generated in response to a demand Memory-write operation counted by DSNP_HIT_WR that hits in a cache outside the local PE cluster on the same device.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

0x819C DSNP_HIT_RW, Snoop hit, demand data access, event

The counter counts each snoop generated in response to a demand Memory-read operation or demand Memory-write operation that hits in a cache outside of the cache hierarchy of this PE.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

`0x819D` DSNP_HIT_NEAR_RW, Snoop hit in near cache, demand data access, event

The counter counts each snoop generated in response to a demand Memory-read operation or demand Memory-write operation counted by DSNP_HIT_RW that hits in a cache outside of the cache hierarchy of this PE in the local PE cluster.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

`0x819E` DSNP_HIT_FAR_RW, Snoop hit in far cache, demand data access, event

The counter counts each snoop generated in response to a demand Memory-read operation or demand Memory-write operation counted by DSNP_HIT_RW that hits in a cache outside the local PE cluster on the same device.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

`0x81B4` DSNP_HIT, Snoop hit, data, event

The counter counts each snoop that hits in a cache outside of the cache hierarchy of this PE.

Related telemetry artifacts

Metrics

- system_peer_cluster_cache_hit_ratio

Metric groups

System_Memory_Effectiveness

Functional groups

Coherency

`0x81B5` DSNP_HIT_NEAR, Snoop hit in near cache, data, event

The counter counts each snoop counted by DSNP_HIT that hits in a cache outside of the cache hierarchy of this PE in the local PE cluster.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

0x81B6 DSNP_HIT_FAR, Snoop hit in far cache, data, event

The counter counts each snoop counted by DSNP_HIT that hits in a cache outside the local PE cluster on the same device.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Coherency

6.19 Events without a functional group for C1-Nano

Events in this section do not have a functional group.

Table 6-19: No functional group events summary

Code	Mnemonic	Name	Description
0x00C3	IMP_L2D_WS_MODE	L2 cache write streaming mode	If the complex is configured with a per-complex L2 cache, this event counts for each cycle where...
0x00C4	IMP_L1D_WS_MODE_ENTRY	L1 data cache entering write streaming mode	This event counts for each entry into write streaming mode.
0x00C5	IMP_L1D_WS_MODE	L1 data cache write streaming mode	This event counts for each cycle where the core is in write streaming mode and is not allocating...
0x00C7	IMP_L3D_WS_MODE	L3 cache write streaming mode	This event counts for each cycle where the core is in write streaming mode and is not allocating...
0x00C8	IMP_LL_WS_MODE	Last level cache write streaming mode	If IMP_CPUECTLR_EL1.EXTLLC is set, this event counts for each cycle where the core is in write...
0x00C9	IMP_L1D_CACHE_WR_NO_ALLOC	Non-allocating L1D_CACHE_WR access	This event counts every memory write operation counted by L1D_CACHE_WR that misses in the cache...
0x00CB	IMP_L2D_WS_MODE_WR_COUNT	L2 cache write streaming mode write	This event counts each L2 memory write operation affected by, at least, IMP_L2D_WS_MODE.
0x00CC	IMP_L3D_WS_MODE_WR_COUNT	L3 cache write streaming mode write	This event counts each L2 memory write operation affected by, at least, IMP_L3D_WS_MODE.
0x00CD	IMP_LL_WS_MODE_WR_COUNT	LL cache write streaming mode write	This event counts each L2 memory write operation affected by IMP_LL_WS_MODE.

Code	Mnemonic	Name	Description
0x00D0	IMP_L2D_WALK_TLB	L2 TLB walk cache access	This event counts each access to the TLB walk cache. This event does not count if the MMU is...
0x00D1	IMP_L2D_WALK_TLB_REFILL	L2 TLB walk cache refill	This event counts each refill of the TLB walk cache. This event does not count if the MMU is...
0x00D4	IMP_L2D_S2_TLB	L2 TLB IPA cache access	This event counts on each access to the IPA cache. If a single translation table walk needs to...
0x00D5	IMP_L2D_S2_TLB_REFILL	L2 TLB IPA cache refill	This event counts on each refill of the IPA cache. If a single translation table walk needs to...
0x00D6	IMP_L2D_CACHE_STASH_DROPPED	L2 cache stash dropped	This event counts on each stash request that is received from the interconnect or the Accelerator...
0x00D7	IMP_L1D_TLB_REFILL_ETS	L1D TLB refill due to ETS replay	This event counts any refill counted by L1D_TLB_REFILL due to ETS replay.
0x00E5	IMP_STALL_BACKEND_ILOCK_ADDR	No operation issued due to the backend, input dependency, address	This event counts every cycle counted by STALL_BACKEND_ILOCK, where there is an input dependency...
0x00ED	IMP_STALL_BACKEND_BUSY_VPU_HAZARD	No operation issued due to the backend, VPU hazard.	This event counts cycles where the core stalls due to contention for the VPU with the other core.
0x00F0	IMP_INST_SPEC_LDST_NUKE	Instruction re-executed, read-after-read hazard	This event counts each instruction which re-executes due to read-after-read hazard

For a complete list of the events in C1-Nano, see PMU events cheat sheet for C1-Nano and PMU events lookup table for C1-Nano.

0x00C3 IMP_L2D_WS_MODE, L2 cache write streaming mode, event

If the complex is configured with a per-complex L2 cache, this event counts for each cycle where the core is in write streaming mode and is not allocating writes into the L2 cache. If the complex is configured without a per-complex L2 cache, this event counts the cluster cache event, as defined by IMP_L3D_WS_MODE. If neither a per-complex cache or a cluster cache is configured, this event is not implemented.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00C4 IMP_L1D_WS_MODE_ENTRY, L1 data cache entering write streaming mode, event

This event counts for each entry into write streaming mode.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00C5 IMP_L1D_WS_MODE, L1 data cache write streaming mode, event

This event counts for each cycle where the core is in write streaming mode and is not allocating writes into the L1 data cache.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00C7 IMP_L3D_WS_MODE, L3 cache write streaming mode, event

This event counts for each cycle where the core is in write streaming mode and is not allocating writes into the L3 cache. If either the complex is configured without a per-complex L2 cache or the cluster is configured without an L3 cache, this event is not implemented.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00C8 IMP_LL_WS_MODE, Last level cache write streaming mode, event

If IMP_CPUECTLR_EL1.EXTLLC is set, this event counts for each cycle where the core is in write streaming mode and is not allocating writes into the system cache. If IMP_CPUECTLR_EL1.EXTLLC is not set, this event is a duplicate of the L*D_WS_MODE event corresponding to the last level of cache implemented in the cluster. That is: - IMP_L3D_WS_MODE, if both per-complex L2 cache and cluster L3 cache are implemented - L2D_WS_MODE, if only one of these caches are implemented - L1D_WS_MODE if neither is implemented.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00C9 IMP_L1D_CACHE_WR_NO_ALLOC, Non-allocating L1D_CACHE_WR access, event

This event counts every memory write operation counted by L1D_CACHE_WR that misses in the cache and does not cause a cache refill.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00CB IMP_L2D_WS_MODE_WR_COUNT, L2 cache write streaming mode write, event

This event counts each L2 memory write operation affected by, at least, IMP_L2D_WS_MODE.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00CC IMP_L3D_WS_MODE_WR_COUNT, L3 cache write streaming mode write, event

This event counts each L2 memory write operation affected by, at least, IMP_L3D_WS_MODE.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00CD IMP_LL_WS_MODE_WR_COUNT, LL cache write streaming mode write, event

This event counts each L2 memory write operation affected by IMP_LL_WS_MODE.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00D0 IMP_L2D_WALK_TLB, L2 TLB walk cache access, event

This event counts each access to the TLB walk cache. This event does not count if the MMU is disabled.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00D1 IMP_L2D_WALK_TLB_REFILL, L2 TLB walk cache refill, event

This event counts each refill of the TLB walk cache. This event does not count if the MMU is disabled.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00D4 IMP_L2D_S2_TLB, L2 TLB IPA cache access, event

This event counts on each access to the IPA cache. If a single translation table walk needs to make multiple accesses to the IPA cache, each access is counted. If stage 2 translation is disabled, this event does not count.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00D5 IMP_L2D_S2_TLB_REFILL, L2 TLB IPA cache refill, event

This event counts on each refill of the IPA cache. If a single translation table walk needs to make multiple accesses to the IPA cache, each access that causes a refill is counted. If stage 2 translation is disabled, this event does not count.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00D6 IMP_L2D_CACHE_STASH_DROPPED, L2 cache stash dropped, event

This event counts on each stash request that is received from the interconnect or the Accelerator Coherency Port (ACP), that targets L2 and is dropped due to lack of buffer space to hold the request. If the core is not configured with a per-complex L2 cache, this event is not implemented.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00D7 IMP_L1D_TLB_REFILL_ETS, L1D TLB refill due to ETS replay, event

This event counts any refill counted by L1D_TLB_REFILL due to ETS replay.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00E5 IMP_STALL_BACKEND_ILOCK_ADDR, No operation issued due to the backend, input dependency, address, event

This event counts every cycle counted by STALL_BACKEND_ILOCK, where there is an input dependency on an address operand. This type of interlock is caused by a load/store instruction waiting for data to calculate the address.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00ED IMP_STALL_BACKEND_BUSY_VPU_HAZARD, No operation issued due to the backend, VPU hazard., event

This event counts cycles where the core stalls due to contention for the VPU with the other core.

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

0x00F0 IMP_INST_SPEC_LDST_NUKE, Instruction re-executed, read-after-read hazard, event

This event counts each instruction which re-executes due to read-after-read hazard

Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in this Telemetry Specification.

Functional groups

Events without a functional group

Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant

export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

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

Revision history

These sections can help you understand how the document has changed over time.

Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

Document history

Issue	Date	Confidentiality	Change
0200-02	10 September 2025	Non-Confidential	Second issue for all revisions of Arm® C1-Nano.
0100-01	1 August 2024	Confidential	First issue for all revisions of Arm® C1-Nano.

Change history

The Change history tables describe the technical changes between released issues of this document in reverse order. Issue numbers match the revision history in Document release information on page 287.

Table 2: Issue 0100-01

Change	Location
First issue for all revisions of Arm® C1-Nano.	-

Table 3: Differences between Issue 0100-01 and 0200-02

Change	Location
Second issue for all revisions of Arm® C1-Nano.	-
Previous release issue numbering and change information realigned for consistency. Corresponding release date information remains unchanged.	-
Clarified introductory text	Overview of the C1-Nano Telemetry methodology
Edited the documentation and resources list	Documentation and resources
Clarifications throughout section	Stage 1: Topdown analysis
Added information about Topdown SME2 metric group	Stage 1: Topdown analysis
New section	Topdown SME2 metric group
Various changes throughout section	Metrics cheat sheet for C1-Nano
Various changes throughout section	PMU events cheat sheet for C1-Nano
Various changes throughout section	Metrics lookup table for C1-Nano
Various changes throughout section	PMU events lookup table for C1-Nano
Added L3 cache MPKI metric	MPKI metrics for C1-Nano
Added L3 cache miss ratio metric	Miss_Ratio metrics for C1-Nano
New section	SVE Effectiveness metric group
New section	FP_Precision_Mix metrics for C1-Nano
New section	Level 3 Cache Effectiveness metric group
New section	System memory effectiveness
Added related telemetry artifacts for L2D_CACHE_REFILL, L2I_CACHE_REFILL, and L2D_CACHE_REFILL_RD	L2_Cache (L2 CACHE) events for C1-Nano
Added related telemetry artifacts for L3D_CACHE_RD, L3D_CACHE_REFILL_RD, and L3D_CACHE_HIT_RD	L3_Cache (L3 CACHE) events for C1-Nano
Added related telemetry artifacts for INST_RETIRED	Retired (RETIRED) events for C1-Nano

Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
bold	Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<and>`	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.

We recommend the following. If you do not follow these recommendations your system might not work.

Your system requires the following. If you do not follow these requirements your system will not work.

You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.

This information is important and needs your attention.

Tip

This information might help you perform a task in an easier, better, or faster way.

Remember

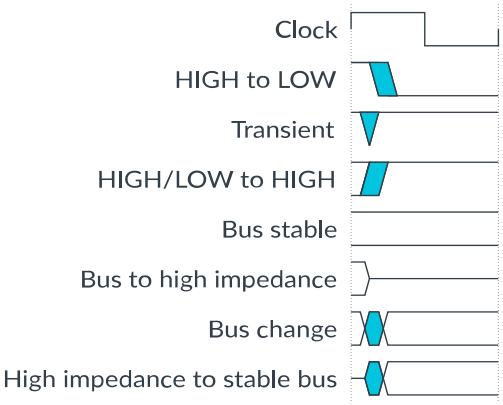
This information reminds you of something important relating to the current content.

Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

Figure 1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lowercase n

At the start or end of a signal name, n denotes an active-LOW signal.

Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Arm documents are available on developer.arm.com/documentation.

Confidential documents are only available to licensees, when logged in. Each document link in the tables below provides direct access to the online version of the document.

Arm product resources	Document ID	Confidentiality
Arm® C1-Nano Core Technical Reference Manual	107753	Non-Confidential
Arm® C1-Scalable Matrix Extension 2 Telemetry Specification	108821	Non-Confidential
Arm® CPU Telemetry Solution Topdown Methodology Specification	109542	Non-Confidential
Arm® Telemetry Solution GitLab repository	–	Non-Confidential
Arm® Telemetry on Arm Developer	–	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Architecture Reference Manual for A-profile architecture	DDI 0487	Non-Confidential