



# Arm® C1-Scalable Matrix Extension 2

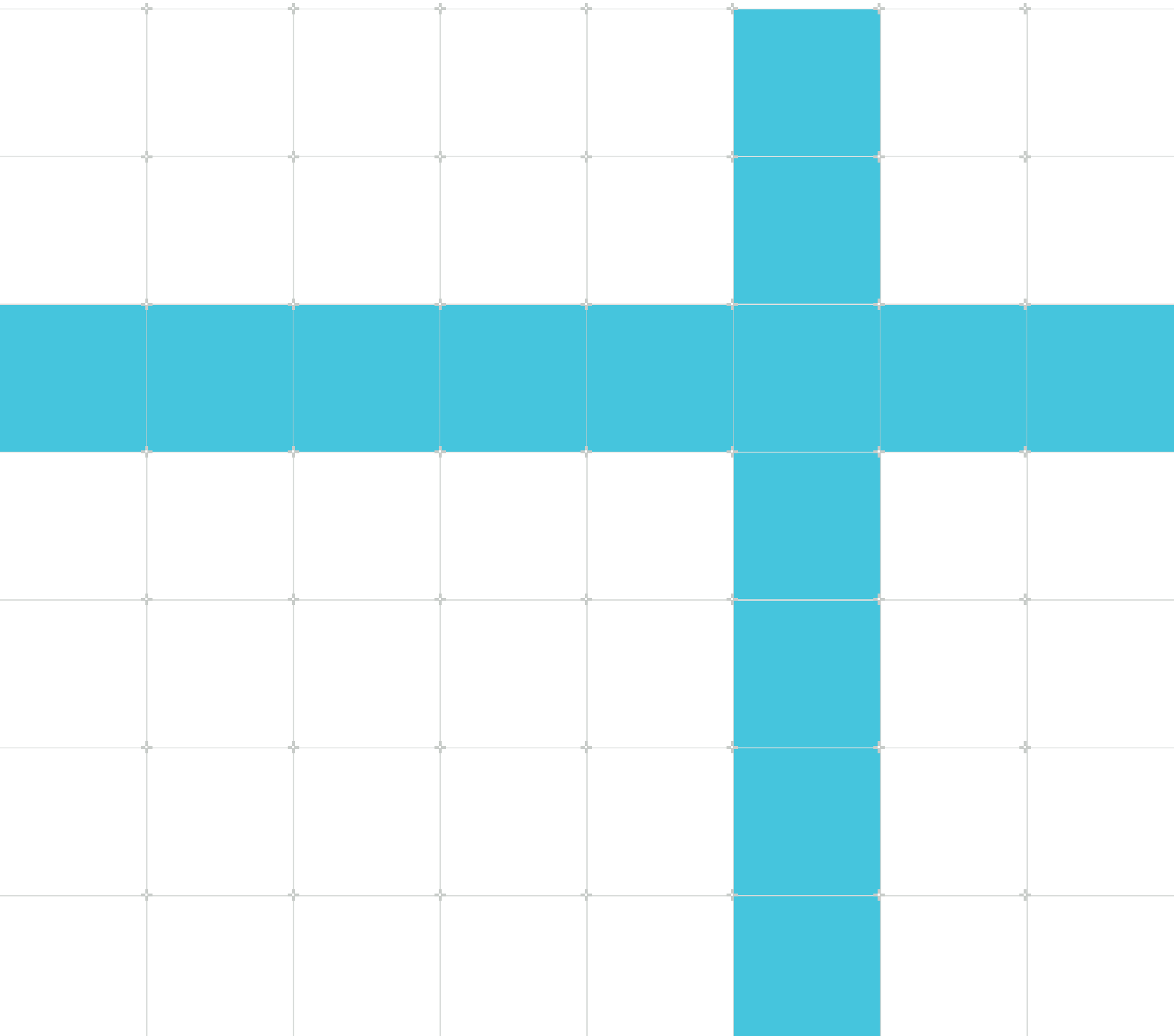
## Telemetry Specification

**Non-Confidential**

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.

**Issue 04**

109821\_0400\_04\_en



# Arm® C1-Scalable Matrix Extension 2 Telemetry Specification

This document is Non-Confidential.

Copyright © 2024–2025 Arm Limited (or its affiliates). All rights reserved.

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

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

This document (109821\_0400\_04\_en) was issued on 2025-09-10. There might be a later issue at <https://developer.arm.com/documentation/109821>

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-SME2 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](mailto: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

<b>1. Overview of the C1-SME2 Telemetry methodology.....</b>	<b>5</b>
1.1 Documentation and resources.....	6
<b>2. Telemetry features of the C1-SME2 unit.....</b>	<b>7</b>
<b>3. C1-SME2 performance analysis methodology.....</b>	<b>9</b>
3.1 Stage 1: Topdown analysis.....	11
3.1.1 SME2 Topdown Level 1 metric group.....	13
3.1.2 SME2 Topdown Frontend metric group.....	14
3.1.3 SME2 Topdown Backend metric group.....	15
3.2 Stage 2: Microarchitecture exploration.....	16
<b>4. C1-SME2 Telemetry cheat-sheets and lookup tables.....</b>	<b>21</b>
4.1 Metrics cheat sheet for C1-SME2.....	21
4.2 PMU events cheat sheet for C1-SME2.....	22
4.3 Metrics lookup table for C1-SME2.....	25
4.4 PMU events lookup table for C1-SME2.....	27
<b>5. Metrics by metric group in C1-SME2.....</b>	<b>39</b>
5.1 CME_Topdown_L1 metrics for C1-SME2.....	39
5.2 CME_Topdown_Frontend metrics for C1-SME2.....	41
5.3 CME_Topdown_Backend metrics for C1-SME2.....	42
5.4 CME_System_Memory_Effectiveness metrics for C1-SME2.....	45
5.5 CME_General metrics for C1-SME2.....	47
5.6 CME_MPKI metrics for C1-SME2.....	48
5.7 CME_Miss_Ratio metrics for C1-SME2.....	49
5.8 CME_L1D_Cache_Effectiveness metrics for C1-SME2.....	52
5.9 CME_L3D_Cache_Effectiveness metrics for C1-SME2.....	54
5.10 CME_Bus_Effectiveness metrics for C1-SME2.....	56
5.11 CME_LL_Cache_Effectiveness metrics for C1-SME2.....	57
5.12 CME_Operation_Mix metrics for C1-SME2.....	59
5.13 CME_IQ_Effectiveness metrics for C1-SME2.....	68
5.14 CME_Port_Utilization metrics for C1-SME2.....	70

5.15 CME_Prefetcher_Effectiveness metrics for C1-SME2.....	73
--	----

## **6. PMU events by functional group in C1-SME2..... 76**

6.1 CME_Bus (SME2 BUS) events for C1-SME2.....	76
6.2 CME_L1D_Cache (SME2 L1D CACHE) events for C1-SME2.....	79
6.3 CME_L3D_Cache (SME2 L3D CACHE) events for C1-SME2.....	89
6.4 CME_LL_Cache (SME2 LL CACHE) events for C1-SME2.....	93
6.5 CME_Memory (SME2 MEMORY) events for C1-SME2.....	95
6.6 CME_Retired (SME2 RETIRED) events for C1-SME2.....	99
6.7 CME_Spec_Operation (SME2 SPEC OPERATION) events for C1-SME2.....	101
6.8 CME_Stall (SME2 STALL) events for C1-SME2.....	124
6.9 CME_General (SME2 GENERAL) events for C1-SME2.....	129
6.10 CME_SVE (SME2 SVE) events for C1-SME2.....	131
6.11 CME_FP_Operation (CME_FP_Operation) events for C1-SME2.....	146
6.12 CME_IQ_Efficiency (SME2 ISSUE QUEUE) events for C1-SME2.....	150
6.13 CME_Port_Utilization (SME2 PORT UTILIZATION) events for C1-SME2.....	152
6.14 CME_Coherency (SME2 COHERENCY) events for C1-SME2.....	154
6.15 CME_System (SME2 SYSTEM) events for C1-SME2.....	155

## **Proprietary notice.....156**

## **Product and document information..... 158**

Product status.....	158
Revision history.....	158
Conventions.....	159

## **Useful resources..... 162**

# 1. Overview of the C1-SME2 Telemetry methodology

The Arm® C1-SME2 *Telemetry Specification* describes the Topdown methodology, derived metrics, and Performance Monitoring Unit (PMU) events supported by the Arm C1-SME2 unit.

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



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-SME2 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-SME2 consist of:

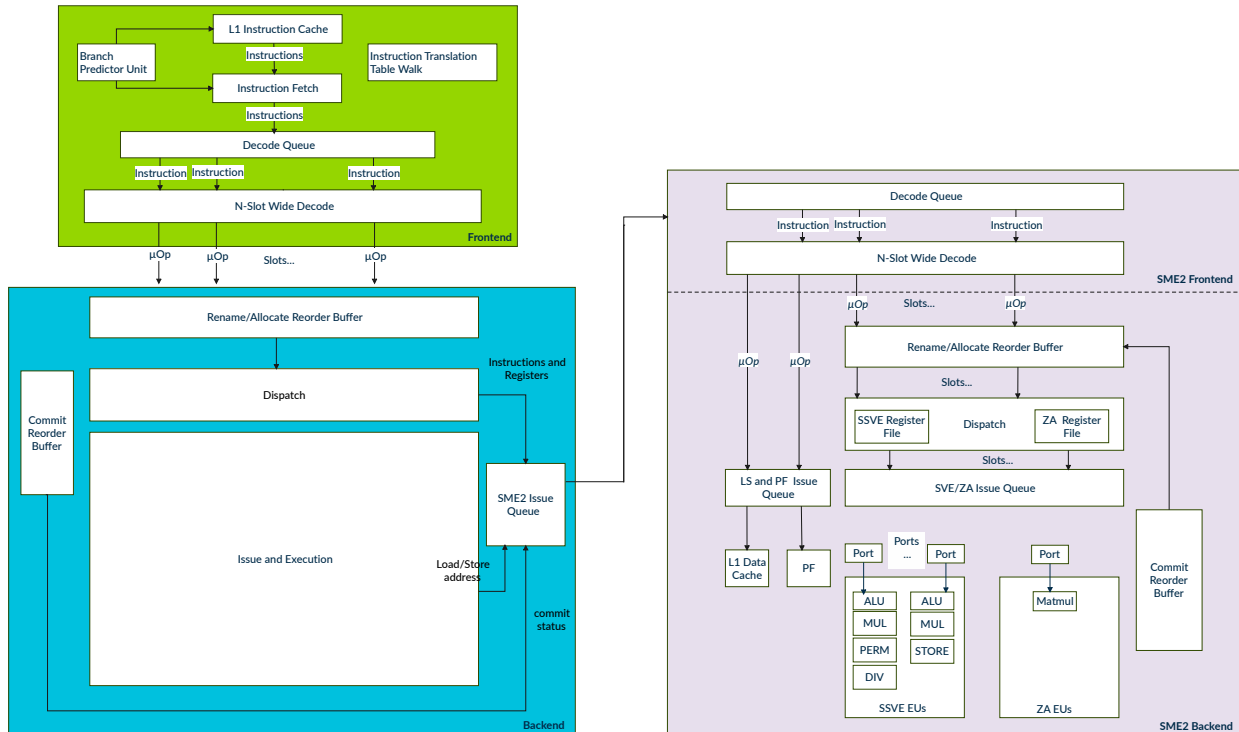
- [\*Arm® Telemetry on Arm Developer\*](#)
- [\*Arm® C1-Scalable Matrix Extension 2 Technical Reference Manual\*](#)

## 2. Telemetry features of the C1-SME2 unit

The C1-SME2 unit is designed as a co-processor to the CPU to execute the Scalable Matrix Extension 2 (SME2) that includes the streaming Scalable Vector Extension (SVE) operations.

The following figure shows the microarchitecture details of the CPU C1-SME2 unit.

**Figure 2-1: C1-SME2 and CPU uarch**



This dedicated co-processing unit has execution pipelines very similar to the CPU backend with a minimal frontend. The frontend of the SME2 unit pipeline contains the decode units for the instructions sent from the CPU.

The CPU is responsible for fetching the instructions from the memory and dispatching the streaming instructions and packets to the SME2 module. All the instructions sent to the SME2 unit must have been committed inside the CPU. During the streaming mode, the CPU is still executing instructions that are not in the SME2 scope, e.g. the instructions don't use vector, predicate, accumulation registers. In addition, all the load/store instructions which are executed in the SME2 unit need the CPU's help to generate the physical address and other control information. The CPU and the SME2 unit support the data transfer between each other:

- The CPU can send register contents to the SME2 unit to update the registers.
- The SME2 unit can also send the data produced inside the SME2 unit back to the CPU

The frontend of the SME2 unit decodes the instructions sent from the CPU and tracks the instructions to the uops to the backend of the SME2 unit. The backend of the SME2 unit renames the registers to resolve the data dependencies and dispatches the instructions to the issue queues where the uops becomes out-of-order. The issue queues send uops to the associated execution units when the uop is qualified to be issued. When the uop is being dispatched, it also gets allocated in the RCQ (register commit queue). When the uops complete the execution, they will be retired in the RCQ in order.

In the dispatch unit, issue queues are employed for:

- Queuing the micro-operations (uops) to assigned ports
- Tracking availability for operands and execution units

Each issue supports multiple categories of operations. Completed operations are retired architecturally in the right program order. The typical operation types supported by different execution units are:

- Vx0: SVE datapath 0
- Vx1: SVE datapath 1
- Matrix Multiplication Unit (Matmul)
- Load pipe
- Store pipe
- Multicycle execution unit: for division and square root instructions

The Memory subsystem of the C1-SME2 unit synchronizes with the CPU for the co-execution. The CPU is responsible for the memory translation and handing the exceptions. The memory operations dispatched to the C1-SME2 unit obtain physical address from the CPU. The C1-SME2 unit has a local L1 cache and supports hardware prefetching. The C1-SME2 unit shares the L3 cache in the DynamiQ™ Shared Unit (DSU) cluster with other CPUs, but does not support a private Level 2 cache, like other CPUs.

## 3. C1-SME2 performance analysis methodology

The Arm Topdown methodology for the C1-SME2 unit 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 SME2 unit, 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

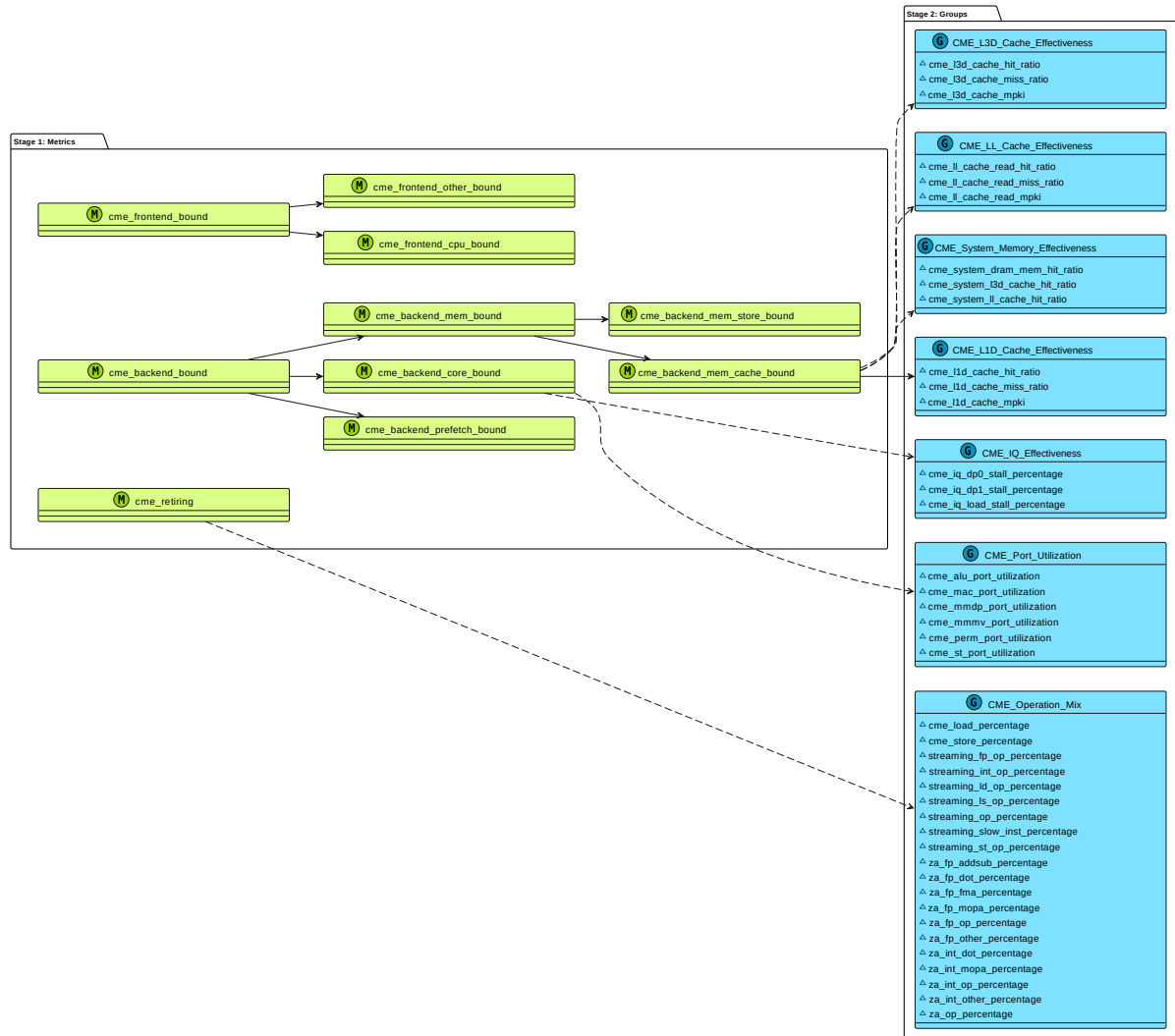
The second stage is to conduct microarchitecture exploration to further analyze bottlenecked SME2 unit resources, based on the Stage 1 findings. Stage 2 uses a set of SME2 resource-effectiveness metrics. For more information, see [Stage 2: Microarchitecture Exploration](#).

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

Arm recommends collecting all metrics that are in Stage 1 and Stage 2 Topdown analysis for workload characterization. Arm provides 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-SME2 unit. 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-SME2**



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

## 3.1 Stage 1: Topdown analysis

The objective of the Topdown analysis is to identify potential bottlenecks in the key blocks of the SME2 unit.

The SME2 unit can be viewed as one of the additional units available, such as the SVE unit and floating point unit, with a new set of registers to support their execution. However, micro-architecturally, the key characteristics of the SME2 unit need to be folded into the CPU execution model, sufficiently enough to conduct performance analysis of the code as well as support software tuning efforts.

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.

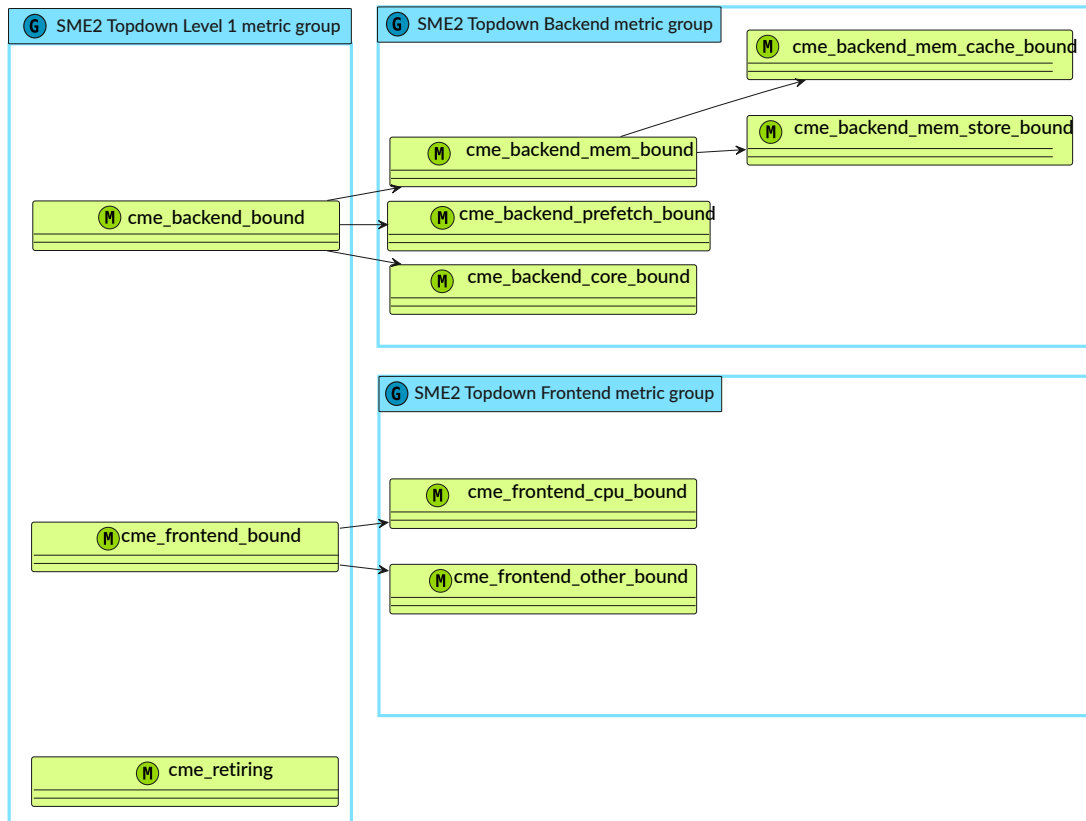


The SME2 unit is a co-processor to the CPU. CPU backend stalls indicate when the SME2 unit is the bottleneck in execution. For example, a high `backend_cme_backpressure_bound` stall in the CPU Topdown SME2 metric group. For more information on the analysis of the SME2 unit execution, see the associated Telemetry Specification for your CPU.

---

The following figure shows the metric groups and metrics in the Stage 1 Topdown methodology tree for C1-SME2, which supports up to four levels of hierarchical pipeline stall accounting.

**Figure 3-2: C1-SME2 Topdown methodology Stage 1 overview**



C1-SME2 has four Stage 1 metric groups.

### SME2 Topdown Level 1

The [SME2 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 [CME\\_Topdown\\_L1](#).

### SME2 Topdown Frontend

The [SME2 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 [CME\\_Topdown\\_Frontend](#).

## SME2 Topdown Backend

The [SME2 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 [CME\\_Topdown\\_Backend](#).

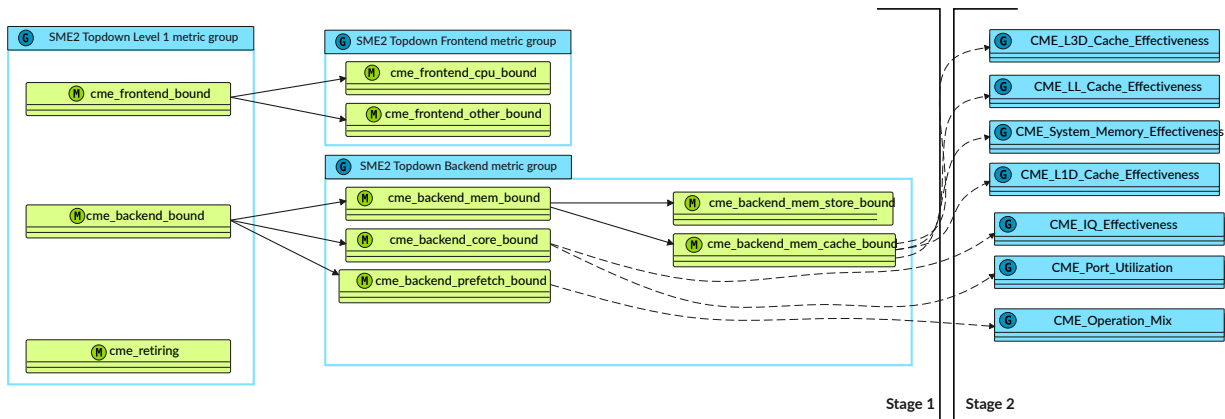
### 3.1.1 SME2 Topdown Level 1 metric group

The [CME\\_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 SME2 Topdown L1 and the next steps in the methodology.

**Figure 3-3: C1-SME2 Metric Group: Topdown\_L1**



The metrics in the SME2 Topdown Level 1 metric group are:

#### **cme\_frontend\_bound**

The [cme\\_frontend\\_bound](#) metric measures pipeline inefficiency due to slots that were stalled because of resource constraints in the frontend.

For further analysis, the next step is the Stage 1 [SME2 Topdown Frontend metric group](#).

#### **cme\_backend\_bound**

The [cme\\_backend\\_bound](#) metric measures pipeline inefficiency due to slots that were stalled because of resource constraints in the backend.

For further analysis, the next step is the Stage 1 [SME2 Topdown Backend metric group](#).

## cme\_retiring

The [cme\\_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 SME2 unit.

## 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 SME2 unit implementation.
- The boundary of the frontend and the backend is the decode and the rename modules.
- A frontend stall is counted when there are no instructions sent to the CPU.
- A backend stall is counted when there are instructions in the post-decode modules, but they cannot be dispatched due to the backend resource constraints.
- In a cycle where there are no instructions in the post-decode modules and instructions are dispatched to the backend, the CPU counts a frontend stall.

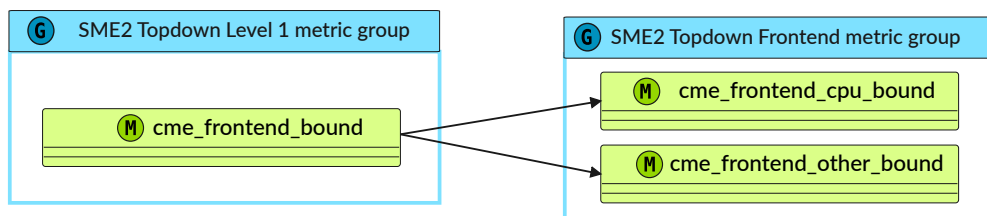
### 3.1.2 SME2 Topdown Frontend metric group

The [CME\\_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 SME2 Topdown Frontend metric group and the next steps in the methodology.

**Figure 3-4: C1-SME2 Metric Group: Topdown\_Frontend**



The metrics in the SME2 Topdown Frontend metric group are:

#### cme\_frontend\_other\_cpu\_bound

The [cme\\_frontend\\_other\\_cpu\\_bound](#) metric is the percentage of total cycle stalled at the frontend due to other CPUs sending fast context switch instructions to the SME2 unit frontend which takes the resources from the current arbitrated CPU.

## cme\_frontend\_cpu\_bound

The [cme\\_frontend\\_cpu\\_bound](#) metric is the percentage of total cycles stalled in the frontend due to frontend SME2 resource constraints not related to instruction fetch latency issues caused by memory access components.

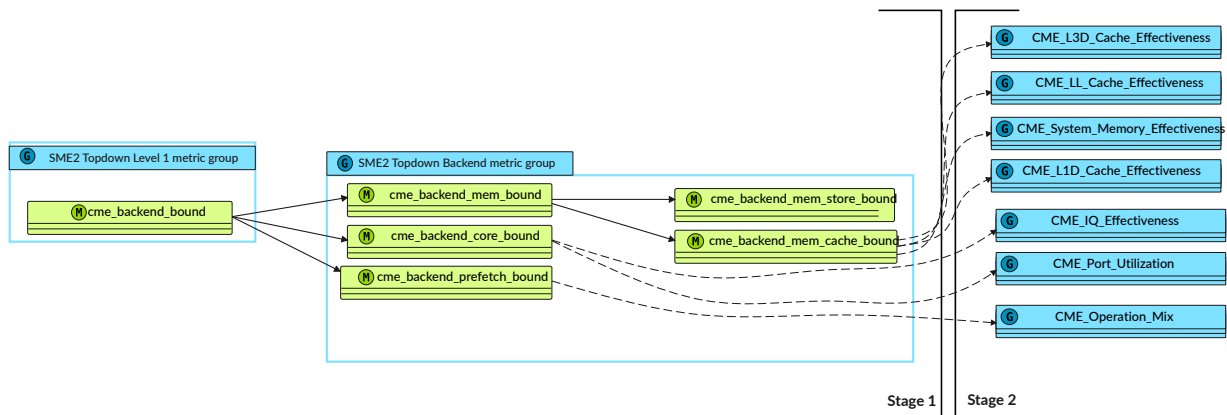
The reason for a stall in the frontend due to the CPU is no instructions were sent from the CPU side to the SME2 unit.

## 3.1.3 SME2 Topdown Backend metric group

The [CME\\_Topdown\\_Backend](#) metric group contains a set of metrics to analyze a backend bound workload.

The following figure shows the metrics for the SME2 Topdown Backend metric group and the next steps in the methodology.

**Figure 3-5: C1-SME2 Metric Group: CME\_Topdown\_Backend**



The metrics in the SME2 Topdown Backend metric group are:

### cme\_backend\_mem\_bound

The [cme\\_backend\\_mem\\_bound](#) metric is the percentage of total cycles stalled in the backend due to Load-Store issue queues not accepting instructions.

### cme\_backend\_core\_bound

The [cme\\_backend\\_core\\_bound](#) metric is the percentage of total cycles stalled in the backend due to DP issue queues not accepting instructions.

### cme\_backend\_prefetch\_bound

The [cme\\_backend\\_prefetch\\_bound](#) metric is the percentage of total cycles stalled in the backend due to hardware prefetcher issue queues not accepting instructions.

### cme\_backend\_mem\_store\_bound

The [cme\\_backend\\_mem\\_store\\_bound](#) is the percentage of total cycles stalled in the backend due to store merge buffer queues not accepting instructions.

### **cme\_backend\_mem\_cache\_bound**

The [cme\\_backend\\_mem\\_cache\\_bound](#) metric is the percentage of total cycles stalled in the backend for memory resource constraints, with the oldest instruction in at least one of the LS issue queues waiting for cache arbitration.

### **Topdown Backend implementation criteria**

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

- The implementation of the Backend Bound metric in the CPU is the sum of the backend core bound, the backend memory bound, and the backend prefetch bound, which is equal to 100% of the backend bound metric.

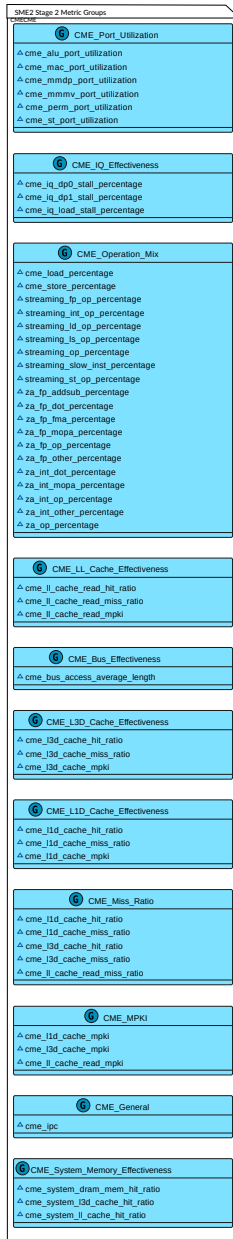
## **3.2 Stage 2: Microarchitecture exploration**

Topdown analysis Stage 2 helps to locate the hot spots in the code and conduct root cause analysis. This stage reveals more details about the SME2 component that caused the bottleneck and was identified during the Stage 1 process.

Stage 2 contains metric groups for microarchitecture exploration targeted at SME2 unit resource effectiveness and other relevant metrics. Metrics in this category are designed for users who are interested in the microarchitectural characteristics of key SME2 unit components. As a co-processor, the SME2 unit is executing instructions with its own front end and backend pipelines that support the execution of streaming instructions/operations dispatched to the SME2 unit. The SME2 unit has most of the pipeline resources like a CPU, including the SME ZA/SVE Z registers that are required for the execution of SME extensions and streaming SVE extensions respectively.

The following figure shows the Stage 2 metric groups for C1-SME2.

**Figure 3-6: C1-SME2 Topdown methodology Stage 2 overview**



In C1-SME2 there are two metric groups based on industry standard metrics, Misses Per Kilo Instructions (MPKI) and Miss Ratios, plus eight other Stage 2 metric groups.

### SME2 Level 1 Data Cache Effectiveness metric group

The [CME\\_L1D\\_Cache\\_Effectiveness](#) metric group contains metrics to evaluate the effectiveness of the L1 Data Cache on the processor. This metric measures the number of SME2 level 1 data cache accesses missed per thousand SME2 instructions executed.

## SME2 Last Level Cache Effectiveness metric group

The [CME\\_LL\\_Cache\\_Effectiveness](#) metric group contains metrics to evaluate the effectiveness of the Last Level Cache on the processor. The SME2 unit has a local L1 cache and supports hardware prefetching. The SME2 unit shares the L3 cache in the DSU cluster with other CPUs, but do not support a private Level 2 cache like other CPU cores.

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

- The [CME\\_LL\\_CACHE\\_RD](#) event counts total SLC accesses made by the core.
- The [CME\\_LL\\_CACHE\\_MISS\\_RD](#) event counts accesses missed at the SLC.

The [cme\\_ll\\_cache\\_read\\_mpki](#) and [cme\\_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 [cme\\_ll\\_cache\\_read\\_hit\\_ratio](#).

Last level cache events do not have a write variant in C1-SME2 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.

## SME2 Level 3 Data Cache Effectiveness metric group

The [CME\\_L3D\\_Cache\\_Effectiveness](#) metric group contains metrics to evaluate the effectiveness of the L3 Data Cache in the DSU.

## SME2 Bus Effectiveness metric group

The [CME\\_Bus\\_Effectiveness](#) metric group contains metrics to evaluate the effectiveness of bus transactions issued by the SME2 unit.

## SME2 Issue Queue Effectiveness metric group

The [CME\\_IQ\\_Effectiveness](#) metric group provides relative stall cycles due to issue queue fullness by operation type.

## SME2 Port Utilization metric group

The [CME\\_Port\\_Utilization](#) metric group provides relative utilization of execution pipes for the program.

## SME2 Operation Mix metric group

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

The SME2 microarchitecture has a variety of executions that can process more than seven types of operations:

- Load/Store Operations executed in the SME2
- Streaming Floating Point Operation

- Streaming Integer Operation
- Streaming Load Operation
- Streaming Store Operation
- Streaming Operation
- Streaming Slow Instruction Operation
- ZA Floating Point addsub operation
- ZA Floating Point dotproduct operation
- ZA Floating Point FMA operation
- ZA Floating Point MOPA operation
- ZA Integer dotproduct operation
- ZA Integer FMA operation
- ZA Integer MOPA operation

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

### SME2 Misses Per Kilo Instructions metric group

The [CME\\_MPKI](#) metric group contains metrics for different resources that can be measured as misses per kilo instructions. These metrics can be used to normalize the misses in SME2 unit components against the total instructions executed. The primary components are L1 cache, L3 Cache and Last-Level Cache.

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



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

---

### SME2 Miss Ratio metric group

The [CME\\_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 SME2 unit components against the total accesses in those components. The primary components are branches, caches, and TLBs L1 cache, L3 Cache and Last-Level Cache.

Miss Ratio metrics provide insights into the efficiency of each component 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.

---

### SME2 General metric group

The [CME\\_General\\_metric\\_group](#) metric group contains the instructions per cycle (IPC) to assess the SME2 performance.

## 4. C1-SME2 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-SME2

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

C1-SME2 specification provides the following types of metrics:

- Total implemented Common metrics: 56

SME2 unit Topdown Level 1 (3)	SME2 unit Topdown Frontend (2)	SME2 unit Topdown Backend (5)
<ul style="list-style-type: none"> <li>cme_backend_bound</li> <li>cme_frontend_bound</li> <li>cme_retiring</li> </ul>	<ul style="list-style-type: none"> <li>cme_frontend_cpu_bound</li> <li>cme_frontend_other_bound</li> </ul>	<ul style="list-style-type: none"> <li>cme_backend_core_bound</li> <li>cme_backend_mem_bound</li> <li>cme_backend_mem_cache_bound</li> <li>cme_backend_mem_store_bound</li> <li>cme_backend_prefetch_bound</li> </ul>
SME2 unit System Memory Effectiveness (3)	SME2 unit General (1)	SME2 unit Misses Per Kilo Instructions (3)
<ul style="list-style-type: none"> <li>cme_system_dram_mem_hit_ratio</li> <li>cme_system_l3d_cache_hit_ratio</li> <li>cme_system_ll_cache_hit_ratio</li> </ul>	<ul style="list-style-type: none"> <li>cme_ipc</li> </ul>	<ul style="list-style-type: none"> <li>cme_l1d_cache_mpki</li> <li>cme_l3d_cache_mpki</li> <li>cme_ll_cache_read_mpki</li> </ul>
SME2 unit Miss Ratio (5)	SME2 unit L1 Data Cache Effectiveness (3)	SME2 unit L3 cluster Cache Effectiveness (3)
<ul style="list-style-type: none"> <li>cme_l1d_cache_hit_ratio</li> <li>cme_l1d_cache_miss_ratio</li> <li>cme_l3d_cache_hit_ratio</li> <li>cme_l3d_cache_miss_ratio</li> <li>cme_ll_cache_read_miss_ratio</li> </ul>	<ul style="list-style-type: none"> <li>cme_l1d_cache_hit_ratio</li> <li>cme_l1d_cache_miss_ratio</li> <li>cme_l1d_cache_mpki</li> </ul>	<ul style="list-style-type: none"> <li>cme_l3d_cache_hit_ratio</li> <li>cme_l3d_cache_miss_ratio</li> <li>cme_l3d_cache_mpki</li> </ul>

SME2 unit Bus Effectiveness (1)	SME2 unit Last Level Cache Effectiveness (3)	SME2 unit Operation Mix (20)
<ul style="list-style-type: none"> <li>cme_bus_access_average_length</li> </ul>	<ul style="list-style-type: none"> <li>cme_ll_cache_read_hit_ratio</li> <li>cme_ll_cache_read_miss_ratio</li> <li>cme_ll_cache_read_mpki</li> </ul>	<ul style="list-style-type: none"> <li>cme_load_percentage</li> <li>cme_store_percentage</li> <li>streaming_fp_op_percentage</li> <li>streaming_int_op_percentage</li> <li>streaming_ld_op_percentage</li> <li>streaming_ls_op_percentage</li> <li>streaming_op_percentage</li> <li>streaming_slow_inst_percentage</li> <li>streaming_st_op_percentage</li> <li>za_fp_addsub_percentage</li> <li>za_fp_dot_percentage</li> <li>za_fp_fma_percentage</li> <li>za_fp_mopa_percentage</li> <li>za_fp_op_percentage</li> <li>za_fp_other_percentage</li> <li>za_int_dot_percentage</li> <li>za_int_mopa_percentage</li> <li>za_int_op_percentage</li> <li>za_int_other_percentage</li> <li>za_op_percentage</li> </ul>
SME2 unit Issue Queue Effectiveness (3)	Execution Unit Effectiveness (6)	SME2 unit Prefetcher Effectiveness (3)
<ul style="list-style-type: none"> <li>cme_iq_dp0_stall_percentage</li> <li>cme_iq_dp1_stall_percentage</li> <li>cme_iq_load_stall_percentage</li> </ul>	<ul style="list-style-type: none"> <li>cme_alu_port_utilization</li> <li>cme_mac_port_utilization</li> <li>cme_mmdp_port_utilization</li> <li>cme_mmmv_port_utilization</li> <li>cme_perm_port_utilization</li> <li>cme_st_port_utilization</li> </ul>	<ul style="list-style-type: none"> <li>cme_l1_prefetcher_accuracy</li> <li>cme_l1_prefetcher_coverage</li> <li>cme_l1_prefetcher_timeliness</li> </ul>

## 4.2 PMU events cheat sheet for C1-SME2

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-SME2 specification provides the following types of PMU events:

- Total implemented Common events: 67
- Total Implemented Product ImpDef events: 145
- PMU Only events : 212
- ETE Only events : 0

SME2 unit Topdown Level 1 (3)	SME2 unit Topdown Frontend (3)	SME2 unit Topdown Backend (6)
<ul style="list-style-type: none"> <li>CME_CYCLES</li> <li>CME_STALL_BACKEND</li> <li>CME_STALL_FRONTEND</li> </ul>	<ul style="list-style-type: none"> <li>CME_STALL_FRONTEND</li> <li>CME_STALL_FRONTEND_CPU</li> <li>CME_STALL_FRONTEND_OTHER_CPU</li> </ul>	<ul style="list-style-type: none"> <li>CME_STALL_BACKEND</li> <li>CME_STALL_BACKEND_CORE</li> <li>CME_STALL_BACKEND_MEM</li> <li>CME_STALL_BACKEND_MEM_CACHE</li> <li>CME_STALL_BACKEND_MEM_STORE</li> <li>CME_STALL_BACKEND_PF</li> </ul>
SME2 unit System Memory Effectiveness (4)	SME2 unit General (2)	SME2 unit Misses Per Kilo Instructions (4)
<ul style="list-style-type: none"> <li>CME_DRAM_ACCESS</li> <li>CME_L1D_CACHE_REFILL</li> <li>CME_L3D_CACHE_HIT</li> <li>CME_LL_CACHE_HIT</li> </ul>	<ul style="list-style-type: none"> <li>CME_CYCLES</li> <li>CME_INST_RETIRED</li> </ul>	<ul style="list-style-type: none"> <li>CME_INST_RETIRED</li> <li>CME_L1D_CACHE_REFILL</li> <li>CME_L3D_CACHE_REFILL</li> <li>CME_LL_CACHE_MISS_RD</li> </ul>
SME2 unit Miss Ratio (8)	SME2 unit L1 Data Cache Effectiveness (4)	SME2 unit L3 cluster Cache Effectiveness (4)
<ul style="list-style-type: none"> <li>CME_L1D_CACHE</li> <li>CME_L1D_CACHE_HIT</li> <li>CME_L1D_CACHE_REFILL</li> <li>CME_L3D_CACHE</li> <li>CME_L3D_CACHE_HIT</li> <li>CME_L3D_CACHE_REFILL</li> <li>CME_LL_CACHE_MISS_RD</li> <li>CME_LL_CACHE_RD</li> </ul>	<ul style="list-style-type: none"> <li>CME_INST_RETIRED</li> <li>CME_L1D_CACHE</li> <li>CME_L1D_CACHE_HIT</li> <li>CME_L1D_CACHE_REFILL</li> </ul>	<ul style="list-style-type: none"> <li>CME_INST_RETIRED</li> <li>CME_L3D_CACHE</li> <li>CME_L3D_CACHE_HIT</li> <li>CME_L3D_CACHE_REFILL</li> </ul>

SME2 unit Bus Effectiveness (2)	SME2 unit Last Level Cache Effectiveness (3)	SME2 unit Operation Mix (22)
<ul style="list-style-type: none"> <li>CME_BUS_ACCESS</li> <li>CME_BUS_REQ</li> </ul>	<ul style="list-style-type: none"> <li>CME_INST_RETIRED</li> <li>CME_LL_CACHE_MISS_RD</li> <li>CME_LL_CACHE_RD</li> </ul>	<ul style="list-style-type: none"> <li>CME_LD_SPEC</li> <li>CME_OP_RETIRED</li> <li>CME_ST_SPEC</li> <li>INST_SPEC</li> <li>SME_FP_ADDSUB_SPEC</li> <li>SME_FP_DOT_SPEC</li> <li>SME_FP_FMA_SPEC</li> <li>SME_FP_MOPA_SPEC</li> <li>SME_FP_OTHER_SPEC</li> <li>SME_FP_SPEC</li> <li>SME_INST_SPEC</li> <li>SME_INT_DOT_SPEC</li> <li>SME_INT_MOPA_SPEC</li> <li>SME_INT_OTHER_SPEC</li> <li>SME_INT_SPEC</li> <li>SSVE_FP_SPEC</li> <li>SSVE_INST_SPEC</li> <li>SSVE_INT_SPEC</li> <li>SSVE_LDST_SPEC</li> <li>SSVE_LD_SPEC</li> <li>SSVE_SLOW_INSTR</li> <li>SSVE_ST_SPEC</li> </ul>
SME2 unit Issue Queue Effectiveness (4)	Execution Unit Effectiveness (7)	SME2 unit Prefetcher Effectiveness (5)
<ul style="list-style-type: none"> <li>CME_DISPATCH_STALL_IQ_DP0</li> <li>CME_DISPATCH_STALL_IQ_DP1</li> <li>CME_DISPATCH_STALL_IQ_LD</li> <li>CME_STALL_BACKEND</li> </ul>	<ul style="list-style-type: none"> <li>CME_CYCLES</li> <li>CME_OP_ALU_ISSUE</li> <li>CME_OP_MAC_ISSUE</li> <li>CME_OP_MMDP_ISSUE</li> <li>CME_OP_MMMV_ISSUE</li> <li>CME_OP_PERM_ISSUE</li> <li>CME_OP_ST_ISSUE</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_CACHE_HIT_RW_FHWPRF</li> <li>CME_L1D_CACHE_REFILL_HWPRF</li> <li>CME_L1D_CACHE_REFILL_RD</li> <li>CME_L1D_CACHE_REFILL_WR</li> <li>CME_L1D_LFB_HIT_RW_FHWPRF</li> </ul>

## 4.3 Metrics lookup table for C1-SME2

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 4-11: Metrics listed by name, with related events and metric groups**

Metric Name	Formula from Events	Metric Groups
cme_alu_port_utilization	$CME\_OP\_ALU\_ISSUE / CME\_CYCLES$	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
cme_backend_bound	$CME\_STALL\_BACKEND / CME\_CYCLES * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_L1</li> </ul>
cme_backend_core_bound	$CME\_STALL\_BACKEND\_CORE / CME\_STALL\_BACKEND * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>
cme_backend_mem_bound	$CME\_STALL\_BACKEND\_MEM / CME\_STALL\_BACKEND * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>
cme_backend_mem_cache_bound	$CME\_STALL\_BACKEND\_MEM\_CACHE / CME\_STALL\_BACKEND\_MEM * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>
cme_backend_mem_store_bound	$CME\_STALL\_BACKEND\_MEM\_STORE / CME\_STALL\_BACKEND\_MEM * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>
cme_backend_prefetch_bound	$CME\_STALL\_BACKEND\_PF / CME\_STALL\_BACKEND * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>
cme_bus_access_average_length	$CME\_BUS\_ACCESS / CME\_BUS\_REQ$	<ul style="list-style-type: none"> <li>CME_Bus_Effectiveness</li> </ul>
cme_frontend_bound	$CME\_STALL\_FRONTEND / CME\_CYCLES * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_L1</li> </ul>
cme_frontend_cpu_bound	$CME\_STALL\_FRONTEND\_CPU / CME\_STALL\_FRONTEND * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_Frontend</li> </ul>
cme_frontend_other_bound	$CME\_STALL\_FRONTEND\_OTHER\_CPU / CME\_STALL\_FRONTEND * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_Frontend</li> </ul>
cme_ipc	$CME\_INST\_RETIRED / CME\_CYCLES$	<ul style="list-style-type: none"> <li>CME_General</li> </ul>
cme_iq_dp0_stall_percentage	$CME\_DISPATCH\_STALL\_IQ\_DP0 / CME\_STALL\_BACKEND * 100$	<ul style="list-style-type: none"> <li>CME_IQ_Effectiveness</li> </ul>
cme_iq_dp1_stall_percentage	$CME\_DISPATCH\_STALL\_IQ\_DP1 / CME\_STALL\_BACKEND * 100$	<ul style="list-style-type: none"> <li>CME_IQ_Effectiveness</li> </ul>
cme_iq_load_stall_percentage	$CME\_DISPATCH\_STALL\_IQ\_LD / CME\_STALL\_BACKEND * 100$	<ul style="list-style-type: none"> <li>CME_IQ_Effectiveness</li> </ul>
cme_l1_prefetcher_accuracy	$(CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF + CME\_L1D\_LFB\_HIT\_RW\_FWWPRF) / CME\_L1D\_CACHE\_REFILL\_HWPRF$	<ul style="list-style-type: none"> <li>CME_Prefetcher_Effectiveness</li> </ul>
cme_l1_prefetcher_coverage	$(CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF + CME\_L1D\_LFB\_HIT\_RW\_FWWPRF) / (CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF + CME\_L1D\_LFB\_HIT\_RW\_FWWPRF + CME\_L1D\_CACHE\_REFILL\_RD + CME\_L1D\_CACHE\_REFILL\_WR)$	<ul style="list-style-type: none"> <li>CME_Prefetcher_Effectiveness</li> </ul>
cme_l1_prefetcher_timeliness	$CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF / (CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF + CME\_L1D\_LFB\_HIT\_RW\_FWWPRF)$	<ul style="list-style-type: none"> <li>CME_Prefetcher_Effectiveness</li> </ul>

Metric Name	Formula from Events	Metric Groups
<ul style="list-style-type: none"> <li>cme_l1d_cache_hit_ratio in CME_L1D_Cache_Effectiveness</li> <li>cme_l1d_cache_hit_ratio in CME_Miss_Ratio</li> </ul>	$\text{CME\_L1D\_CACHE\_HIT} / \text{CME\_L1D\_CACHE}$	<ul style="list-style-type: none"> <li>CME_L1D_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> </ul>
<ul style="list-style-type: none"> <li>cme_l1d_cache_miss_ratio in CME_L1D_Cache_Effectiveness</li> <li>cme_l1d_cache_miss_ratio in CME_Miss_Ratio</li> </ul>	$\text{CME\_L1D\_CACHE\_REFILL} / \text{CME\_L1D\_CACHE}$	<ul style="list-style-type: none"> <li>CME_L1D_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> </ul>
<ul style="list-style-type: none"> <li>cme_l1d_cache_mpki in CME_L1D_Cache_Effectiveness</li> <li>cme_l1d_cache_mpki in CME_MPPI</li> </ul>	$\text{CME\_L1D\_CACHE\_REFILL} / \text{CME\_INST\_RETIRED} * 1000$	<ul style="list-style-type: none"> <li>CME_L1D_Cache_Effectiveness</li> <li>CME_MPPI</li> </ul>
<ul style="list-style-type: none"> <li>cme_l3d_cache_hit_ratio in CME_L3D_Cache_Effectiveness</li> <li>cme_l3d_cache_hit_ratio in CME_Miss_Ratio</li> </ul>	$\text{CME\_L3D\_CACHE\_HIT} / \text{CME\_L3D\_CACHE}$	<ul style="list-style-type: none"> <li>CME_L3D_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> </ul>
<ul style="list-style-type: none"> <li>cme_l3d_cache_miss_ratio in CME_L3D_Cache_Effectiveness</li> <li>cme_l3d_cache_miss_ratio in CME_Miss_Ratio</li> </ul>	$\text{CME\_L3D\_CACHE\_REFILL} / \text{CME\_L3D\_CACHE}$	<ul style="list-style-type: none"> <li>CME_L3D_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> </ul>
<ul style="list-style-type: none"> <li>cme_l3d_cache_mpki in CME_L3D_Cache_Effectiveness</li> <li>cme_l3d_cache_mpki in CME_MPPI</li> </ul>	$\text{CME\_L3D\_CACHE\_REFILL} / \text{CME\_INST\_RETIRED} * 1000$	<ul style="list-style-type: none"> <li>CME_L3D_Cache_Effectiveness</li> <li>CME_MPPI</li> </ul>
cme_ll_cache_read_hit_ratio	$(\text{CME\_LL\_CACHE\_RD} - \text{CME\_LL\_CACHE\_MISS\_RD}) / \text{CME\_LL\_CACHE\_RD}$	<ul style="list-style-type: none"> <li>CME_LL_Cache_Effectiveness</li> </ul>
<ul style="list-style-type: none"> <li>cme_ll_cache_read_miss_ratio in CME_LL_Cache_Effectiveness</li> <li>cme_ll_cache_read_miss_ratio in CME_Miss_Ratio</li> </ul>	$\text{CME\_LL\_CACHE\_MISS\_RD} / \text{CME\_LL\_CACHE\_RD}$	<ul style="list-style-type: none"> <li>CME_LL_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> </ul>
<ul style="list-style-type: none"> <li>cme_ll_cache_read_mpki in CME_LL_Cache_Effectiveness</li> <li>cme_ll_cache_read_mpki in CME_MPPI</li> </ul>	$\text{CME\_LL\_CACHE\_MISS\_RD} / \text{CME\_INST\_RETIRED} * 1000$	<ul style="list-style-type: none"> <li>CME_LL_Cache_Effectiveness</li> <li>CME_MPPI</li> </ul>
cme_load_percentage	$\text{CME\_LD\_SPEC} / \text{CME\_OP\_RETIRED} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
cme_mac_port_utilization	$\text{CME\_OP\_MAC\_ISSUE} / \text{CME\_CYCLES}$	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
cme_mmdp_port_utilization	$\text{CME\_OP\_MMDP\_ISSUE} / \text{CME\_CYCLES}$	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
cme_mmmv_port_utilization	$\text{CME\_OP\_MMMV\_ISSUE} / \text{CME\_CYCLES}$	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
cme_perm_port_utilization	$\text{CME\_OP\_PERM\_ISSUE} / \text{CME\_CYCLES}$	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
cme_retiring	$(\text{CME\_CYCLES} - \text{CME\_STALL\_FRONTEND} - \text{CME\_STALL\_BACKEND}) / \text{CME\_CYCLES} * 100$	<ul style="list-style-type: none"> <li>CME_Topdown_L1</li> </ul>
cme_st_port_utilization	$\text{CME\_OP\_ST\_ISSUE} / \text{CME\_CYCLES}$	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
cme_store_percentage	$\text{CME\_ST\_SPEC} / \text{CME\_OP\_RETIRED} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
cme_system_dram_mem_hit_ratio	$\text{CME\_DRAM\_ACCESS} / \text{CME\_L1D\_CACHE\_REFILL}$	<ul style="list-style-type: none"> <li>CME_System_Memory_Effectiveness</li> </ul>
cme_system_l3d_cache_hit_ratio	$\text{CME\_L3D\_CACHE\_HIT} / \text{CME\_L1D\_CACHE\_REFILL}$	<ul style="list-style-type: none"> <li>CME_System_Memory_Effectiveness</li> </ul>

Metric Name	Formula from Events	Metric Groups
cme_system_ll_cache_hit_ratio	$\text{CME\_LL\_CACHE\_HIT} / \text{CME\_L1D\_CACHE\_REFILL}$	<ul style="list-style-type: none"> <li>CME_System_Memory_Effectiveness</li> </ul>
streaming_fp_op_percentage	$\text{SSVE\_FP\_SPEC} / \text{SSVE\_INST\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
streaming_int_op_percentage	$\text{SSVE\_INT\_SPEC} / \text{SSVE\_INST\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
streaming_ld_op_percentage	$\text{SSVE\_LD\_SPEC} / \text{CME\_OP\_RETIRED} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
streaming_ls_op_percentage	$\text{SSVE\_LDST\_SPEC} / \text{CME\_OP\_RETIRED} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
streaming_op_percentage	$\text{SSVE\_INST\_SPEC} / \text{INST\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
streaming_slow_inst_percentage	$\text{SSVE\_SLOW\_INSTR} / \text{SSVE\_INST\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
streaming_st_op_percentage	$\text{SSVE\_ST\_SPEC} / \text{CME\_OP\_RETIRED} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_fp_addsub_percentage	$\text{SME\_FP\_ADDSUB\_SPEC} / \text{SME\_FP\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_fp_dot_percentage	$\text{SME\_FP\_DOT\_SPEC} / \text{SME\_FP\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_fp_fma_percentage	$\text{SME\_FP\_FMA\_SPEC} / \text{SME\_FP\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_fp_mopa_percentage	$\text{SME\_FP\_MOPA\_SPEC} / \text{SME\_FP\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_fp_op_percentage	$\text{SME\_FP\_SPEC} / \text{SME\_INST\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_fp_other_percentage	$\text{SME\_FP\_OTHER\_SPEC} / \text{SME\_FP\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_int_dot_percentage	$\text{SME\_INT\_DOT\_SPEC} / \text{SME\_INT\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_int_mopa_percentage	$\text{SME\_INT\_MOPA\_SPEC} / \text{SME\_INT\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_int_op_percentage	$\text{SME\_INT\_SPEC} / \text{SME\_INST\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_int_other_percentage	$\text{SME\_INT\_OTHER\_SPEC} / \text{SME\_INT\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>
za_op_percentage	$\text{SME\_INST\_SPEC} / \text{INST\_SPEC} * 100$	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>

## 4.4 PMU events lookup table for C1-SME2

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: 309
- Total implemented Common events: 67
  - Common : Architectural-defined events: 67
  - Common : Implementation-defined events: 0
- Total Implemented Product ImpDef events: 145
- PMU Only Events : 212

- ETE Only Events : 0

**Table 4-12: Events listed by Event Code, with related Metrics, Metric Groups, and Functional Groups**

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x1b, INST_SPEC	<ul style="list-style-type: none"> <li>streaming_op_percentage</li> <li>za_op_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3219, SSVE_INST_SPEC	<ul style="list-style-type: none"> <li>streaming_op_percentage</li> <li>streaming_int_op_percentage</li> <li>streaming_fp_op_percentage</li> <li>streaming_slow_inst_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x321a, SSVE_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x321b, SSVE_LDST_SPEC	<ul style="list-style-type: none"> <li>streaming_ls_op_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x321c, SSVE_LD_SPEC	<ul style="list-style-type: none"> <li>streaming_ld_op_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x321d, SSVE_ST_SPEC	<ul style="list-style-type: none"> <li>streaming_st_op_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x321f, SSVE_INT_SPEC	<ul style="list-style-type: none"> <li>streaming_int_op_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3220, SSVE_FP_SPEC	<ul style="list-style-type: none"> <li>streaming_fp_op_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x3221, SSVE_INT8_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3222, SSVE_INT16_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3223, SSVE_INT32_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3224, SSVE_INT64_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3225, SSVE_FP_HP_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x3226, SSVE_FP_BF16_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x3227, SSVE_FP_SP_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x3228, SSVE_FP_DP_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x3229, SSVE_INT_MUL_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x322a, SSVE_INT_DOT_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x322b, SSVE_FP_ADDSUB_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x322c, SSVE_FP_MUL_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x322d, SSVE_FP_FMA_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x322e, SSVE_FP_DOT_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x322f, SSVE_FP_SQRT_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3230, SSVE_FP_DIV_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3231, SSVE_FP_RECPE_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3232, SSVE_FP_CVT_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3233, SSVE_FP_VREDUCE_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3234, SSVE_PRED_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x3235, SSVE_PRED_EMPTY_SPEC	-	-	• CME_SVE
0x3236, SSVE_PRED_FULL_SPEC	-	-	• CME_SVE
0x3237, SSVE_PRED_NOT_FULL_SPEC	-	-	• CME_SVE
0x3238, SSVE_PRED_PARTIAL_SPEC	-	-	• CME_SVE
0x3239, SSVE_FP_SCALE_OPS_SPEC	-	-	• CME_SVE
0x323a, SSVE_FP_HP_SCALE_OPS _SPEC	-	-	• CME_SVE
0x323b, SSVE_FP_BF16_SCALE_OPS _SPEC	-	-	• CME_SVE
0x323c, SSVE_FP_SP_SCALE_OPS_SPEC	-	-	• CME_SVE
0x323d, SSVE_FP_DP_SCALE_OPS _SPEC	-	-	• CME_SVE
0x323e, SSVE_INT_SCALE_OPS _SPEC	-	-	• CME_SVE
0x323f, SSVE_LDST_SCALE_OPS_SPEC	-	-	• CME_SVE
0x3240, SSVE_LD_SCALE_OPS_SPEC	-	-	• CME_SVE
0x3241, SSVE_ST_SCALE_OPS_SPEC	-	-	• CME_SVE
0x3242, SSVE_LDST_SCALE_BYTES _SPEC	-	-	• CME_SVE
0x3243, SSVE_LD_SCALE_BYTES_SPEC	-	-	• CME_SVE
0x3244, SSVE_ST_SCALE_BYTES_SPEC	-	-	• CME_SVE
0x3245, SSVE_LDST_FIXED_BYTES _SPEC	-	-	• CME_SVE

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x3246, CME_CYCLES	<ul style="list-style-type: none"> <li>cme_retiring</li> <li>cme_frontend_bound</li> <li>cme_backend_bound</li> <li>cme_ipc</li> <li>cme_alu_port_utilization</li> <li>cme_mac_port_utilization</li> <li>cme_mmdp_port_utilization</li> <li>cme_mmmv_port_utilization</li> <li>cme_perm_port_utilization</li> <li>cme_st_port_utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_General</li> <li>CME_Port_Utilization</li> <li>CME_Topdown_L1</li> </ul>	<ul style="list-style-type: none"> <li>CME_General</li> </ul>
0x3247, CME_INST_RETIRED	<ul style="list-style-type: none"> <li>cme_ipc</li> <li>cme_l1d_cache_mpki in CME_L1D_Cache_Effectiveness</li> <li>cme_l1d_cache_mpki in CME_MPKI</li> <li>cme_l3d_cache_mpki in CME_L3D_Cache_Effectiveness</li> <li>cme_l3d_cache_mpki in CME_MPKI</li> <li>cme_ll_cache_read_mpki in CME_LL_Cache_Effectiveness</li> <li>cme_ll_cache_read_mpki in CME_MPKI</li> </ul>	<ul style="list-style-type: none"> <li>CME_General</li> <li>CME_L1D_Cache_Effectiveness</li> <li>CME_L3D_Cache_Effectiveness</li> <li>CME_LL_Cache_Effectiveness</li> <li>CME_MPKI</li> </ul>	<ul style="list-style-type: none"> <li>CME_Retired</li> </ul>
0x3248, CME_OP_RETIRED	<ul style="list-style-type: none"> <li>cme_load_percentage</li> <li>cme_store_percentage</li> <li>streaming_ld_op_percentage</li> <li>streaming_st_op_percentage</li> <li>streaming_ls_op_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Retired</li> </ul>
0x3249, CME_STALL	-	-	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x324a, CME_STALL_FRONTEND	<ul style="list-style-type: none"> <li>cme_retiring</li> <li>cme_frontend_bound</li> <li>cme_frontend_cpu_bound</li> <li>cme_frontend_other_bound</li> </ul>	<ul style="list-style-type: none"> <li>CME_Topdown_Frontend</li> <li>CME_Topdown_L1</li> </ul>	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x324b, CME_STALL_FRONTEND_CPU	<ul style="list-style-type: none"> <li>cme_frontend_cpu_bound</li> </ul>	<ul style="list-style-type: none"> <li>CME_Topdown_Frontend</li> </ul>	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x324c, CME_STALL_FRONTEND_OTHER_CPU	<ul style="list-style-type: none"> <li>cme_frontend_other_bound</li> </ul>	<ul style="list-style-type: none"> <li>CME_Topdown_Frontend</li> </ul>	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x324d, CME_STALL_BACKEND	<ul style="list-style-type: none"> <li>cme_retiring</li> <li>cme_backend_bound</li> <li>cme_backend_core_bound</li> <li>cme_backend_mem_bound</li> <li>cme_backend_prefetch_bound</li> <li>cme_iq_dp0_stall_percentage</li> <li>cme_iq_dp1_stall_percentage</li> <li>cme_iq_load_stall_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_IQ_Effectiveness</li> <li>CME_Topdown_Backend</li> <li>CME_Topdown_L1</li> </ul>	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x324e, CME_STALL_BACKEND_CORE	<ul style="list-style-type: none"> <li>cme_backend_core_bound</li> </ul>	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x324f, CME_STALL_BACKEND_MEM	<ul style="list-style-type: none"> <li>cme_backend_mem_bound</li> <li>cme_backend_mem_cache_bound</li> <li>cme_backend_mem_store_bound</li> </ul>	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x3250, CME_STALL_BACKEND_PF	<ul style="list-style-type: none"> <li>cme_backend_prefetch_bound</li> </ul>	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x3251, CME_STALL_BACKEND_MEM_CACHE	<ul style="list-style-type: none"> <li>cme_backend_mem_cache_bound</li> </ul>	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x3252, CME_STALL_BACKEND_MEM_STORE	<ul style="list-style-type: none"> <li>cme_backend_mem_store_bound</li> </ul>	<ul style="list-style-type: none"> <li>CME_Topdown_Backend</li> </ul>	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x3253, CME_LDST_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3254, CME_LD_SPEC	<ul style="list-style-type: none"> <li>cme_load_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3255, CME_UNALIGNED_LDST_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3256, CME_LDST_ALIGN_LAT	-	-	<ul style="list-style-type: none"> <li>CME_Memory</li> </ul>
0x3257, CME_UNALIGNED_LD_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x3258, CME_LD_ALIGN_LAT	-	-	<ul style="list-style-type: none"> <li>CME_Memory</li> </ul>
0x3259, CME_ST_SPEC	<ul style="list-style-type: none"> <li>cme_store_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x325a, CME_UNALIGNED_ST_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x325b, CME_ST_ALIGN_LAT	-	-	<ul style="list-style-type: none"> <li>CME_Memory</li> </ul>
0x325c, CME_PRF_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x325d, CME_DISPATCH_STALL_IQ_DP0	<ul style="list-style-type: none"> <li>cme_iq_dp0_stall_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_IQ_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_IQ_Efficiency</li> </ul>
0x325e, CME_DISPATCH_STALL_IQ_DP1	<ul style="list-style-type: none"> <li>cme_iq_dp1_stall_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_IQ_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_IQ_Efficiency</li> </ul>

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x325f, CME_DISPATCH_STALL_IQ_LD	<ul style="list-style-type: none"> <li>cme_iq_load_stall_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_IQ_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_IQ_Efficiency</li> </ul>
0x3260, CME_OP_ALU_ISSUE	<ul style="list-style-type: none"> <li>cme_alu_port_utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
0x3261, CME_OP_MAC_ISSUE	<ul style="list-style-type: none"> <li>cme_mac_port_utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
0x3262, CME_OP_PERM_ISSUE	<ul style="list-style-type: none"> <li>cme_perm_port_utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
0x3263, CME_OP_ST_ISSUE	<ul style="list-style-type: none"> <li>cme_st_port_utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
0x3264, CME_OP_MMDP_ISSUE	<ul style="list-style-type: none"> <li>cme_mmdp_port_utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
0x3265, CME_OP_MMMV_ISSUE	<ul style="list-style-type: none"> <li>cme_mmmv_port_utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
0x3266, CME_BUS_REQ_RD_PERCYC	-	-	<ul style="list-style-type: none"> <li>CME_Bus</li> </ul>
0x3267, CME_BUS_REQ	<ul style="list-style-type: none"> <li>cme_bus_access_average_length</li> </ul>	<ul style="list-style-type: none"> <li>CME_Bus_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_Bus</li> </ul>
0x3268, CME_BUS_REQ_RD	-	-	<ul style="list-style-type: none"> <li>CME_Bus</li> </ul>
0x3269, CME_BUS_REQ_WR	-	-	<ul style="list-style-type: none"> <li>CME_Bus</li> </ul>
0x326a, CME_BUS_CYCLES	-	-	<ul style="list-style-type: none"> <li>CME_Bus</li> </ul>
0x326b, CME_BUS_ACCESS	<ul style="list-style-type: none"> <li>cme_bus_access_average_length</li> </ul>	<ul style="list-style-type: none"> <li>CME_Bus_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_Bus</li> </ul>
0x326c, CME_BUS_ACCESS_RD	-	-	<ul style="list-style-type: none"> <li>CME_Bus</li> </ul>
0x326d, CME_BUS_ACCESS_WR	-	-	<ul style="list-style-type: none"> <li>CME_Bus</li> </ul>
0x326e, CME_DSNP_HIT	-	-	<ul style="list-style-type: none"> <li>CME_Coherency</li> </ul>
0x326f, CME_L1D_CACHE	<ul style="list-style-type: none"> <li>cme_l1d_cache_miss_ratio in CME_L1D_Cache_Effectiveness</li> <li>cme_l1d_cache_miss_ratio in CME_Miss_Ratio</li> <li>cme_l1d_cache_hit_ratio in CME_L1D_Cache_Effectiveness</li> <li>cme_l1d_cache_hit_ratio in CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3270, CME_L1D_CACHE_HIT	<ul style="list-style-type: none"> <li>cme_l1d_cache_hit_ratio in CME_L1D_Cache_Effectiveness</li> <li>cme_l1d_cache_hit_ratio in CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3271, CME_L1D_CACHE_HIT_RW	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3272, CME_L1D_CACHE_HIT_RD	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3273, CME_L1D_CACHE_HIT_WR	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x3274, CME_L1D_CACHE_HIT_RW_FHWPRF	<ul style="list-style-type: none"> <li>cme_l1_prefetcher_accuracy</li> <li>cme_l1_prefetcher_timeliness</li> <li>cme_l1_prefetcher_coverage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Prefetcher_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3275, CME_L1D_CACHE_MISS	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3276, CME_L1D_CACHE_RW	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3277, CME_L1D_CACHE_RD	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3278, CME_L1D_CACHE_WR	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3279, CME_L1D_CACHE_PRF	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x327a, CME_L1D_CACHE_HWPRF	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x327b, CME_L1D_CACHE_PRFM	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x327c, CME_L1D_CACHE_REFILL	<ul style="list-style-type: none"> <li>cme_l1d_cache_mпки in CME_L1D_Cache_Effectiveness</li> <li>cme_l1d_cache_mпки in CME_MPкI</li> <li>cme_l1d_cache_miss_ratio in CME_L1D_Cache_Effectiveness</li> <li>cme_l1d_cache_miss_ratio in CME_Miss_Ratio</li> <li>cme_system_dram_mem_hit_ratio</li> <li>cme_system_l3d_cache_hit_ratio</li> <li>cme_system_ll_cache_hit_ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache_Effectiveness</li> <li>CME_MPкI</li> <li>CME_Miss_Ratio</li> <li>CME_System_Memory_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x327d, CME_L1D_CACHE_REFILL_INNER	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x327e, CME_L1D_CACHE_REFILL_OUTER	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3280, CME_L1D_CACHE_REFILL_RD	<ul style="list-style-type: none"> <li>cme_l1_prefetcher_coverage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Prefetcher_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3281, CME_L1D_CACHE_REFILL_WR	<ul style="list-style-type: none"> <li>cme_l1_prefetcher_coverage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Prefetcher_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3282, CME_L1D_CACHE_INVALID	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3283, CME_L1D_CACHE_LMISS_RD	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x3284, CME_L1D_CACHE_WB	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3285, CME_L1D_CACHE_WB_CLEAN	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3286, CME_L1D_CACHE_WB_VICTIM	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x3287, CME_L3D_CACHE	<ul style="list-style-type: none"> <li>cme_l3d_cache_miss_ratio in CME_L3D_Cache_Effectiveness</li> <li>cme_l3d_cache_miss_ratio in CME_Miss_Ratio</li> <li>cme_l3d_cache_hit_ratio in CME_L3D_Cache_Effectiveness</li> <li>cme_l3d_cache_hit_ratio in CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L3D_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x3288, CME_L3D_CACHE_RW	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x3289, CME_L3D_CACHE_RD	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x328a, CME_L3D_CACHE_WR	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x328b, CME_L3D_CACHE_ALLOCATE	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x328d, CME_L3D_CACHE_LMISS_RD	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x328e, CME_L3D_CACHE_HIT	<ul style="list-style-type: none"> <li>cme_l3d_cache_hit_ratio in CME_L3D_Cache_Effectiveness</li> <li>cme_l3d_cache_hit_ratio in CME_Miss_Ratio</li> <li>cme_system_l3d_cache_hit_ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L3D_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> <li>CME_System_Memory_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x328f, CME_L3D_CACHE_MISS	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x3290, CME_L3D_CACHE_REFILL	<ul style="list-style-type: none"> <li>cme_l3d_cache_mпки in CME_L3D_Cache_Effectiveness</li> <li>cme_l3d_cache_mпки in CME_MPKI</li> <li>cme_l3d_cache_miss_ratio in CME_L3D_Cache_Effectiveness</li> <li>cme_l3d_cache_miss_ratio in CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L3D_Cache_Effectiveness</li> <li>CME_MPKI</li> <li>CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x3291, CME_L3D_CACHE_REFILL_PRFM	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x3292, CME_L3D_CACHE_REFILL_RD	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x3293, CME_LL_CACHE	-	-	<ul style="list-style-type: none"> <li>CME_LL_Cache</li> </ul>
0x3294, CME_LL_CACHE_RD	<ul style="list-style-type: none"> <li>cme_ll_cache_read_miss_ratio in CME_LL_Cache_Effectiveness</li> <li>cme_ll_cache_read_miss_ratio in CME_Miss_Ratio</li> <li>cme_ll_cache_read_hit_ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_LL_Cache_Effectiveness</li> <li>CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_LL_Cache</li> </ul>
0x3295, CME_LL_CACHE_MISS_RD	<ul style="list-style-type: none"> <li>cme_ll_cache_read_mпки in CME_LL_Cache_Effectiveness</li> <li>cme_ll_cache_read_mпки in CME_MPPI</li> <li>cme_ll_cache_read_miss_ratio in CME_LL_Cache_Effectiveness</li> <li>cme_ll_cache_read_miss_ratio in CME_Miss_Ratio</li> <li>cme_ll_cache_read_hit_ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_LL_Cache_Effectiveness</li> <li>CME_MPPI</li> <li>CME_Miss_Ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_LL_Cache</li> </ul>
0x3296, CME_LL_CACHE_HIT	<ul style="list-style-type: none"> <li>cme_system_ll_cache_hit_ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_System_Memory_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_LL_Cache</li> </ul>
0x3298, CME_MEM_ACCESS	-	-	<ul style="list-style-type: none"> <li>CME_Memory</li> </ul>
0x3299, CME_MEM_ACCESS_RD	-	-	<ul style="list-style-type: none"> <li>CME_Memory</li> </ul>
0x329a, CME_MEM_ACCESS_RD_PERCYC	-	-	<ul style="list-style-type: none"> <li>CME_Memory</li> </ul>
0x329b, CME_MEM_ACCESS_WR	-	-	<ul style="list-style-type: none"> <li>CME_Memory</li> </ul>
0x329c, CME_REMOTE_ACCESS	-	-	<ul style="list-style-type: none"> <li>CME_Memory</li> </ul>
0x329d, CME_REMOTE_ACCESS_RD	-	-	<ul style="list-style-type: none"> <li>CME_Memory</li> </ul>
0x32a4, SSVE_SLOW_INSTR	<ul style="list-style-type: none"> <li>streaming_slow_inst_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_General</li> </ul>
0x32a5, SSVE_GPR_UPDATE	-	-	<ul style="list-style-type: none"> <li>CME_General</li> </ul>
0x32a6, SSVE_PRED_UPDATE	-	-	<ul style="list-style-type: none"> <li>CME_General</li> </ul>
0x32a7, SSVE_FLAG_UPDATE	-	-	<ul style="list-style-type: none"> <li>CME_General</li> </ul>
0x32a8, SSVE_CONTEXT_SWITCH	-	-	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x32a9, SSVE_FAST_CONTEXT_SWITCH	-	-	<ul style="list-style-type: none"> <li>CME_Stall</li> </ul>
0x32aa, CME_OP_DIVSQRT_ISSUE	-	-	<ul style="list-style-type: none"> <li>CME_Port_Utilization</li> </ul>
0x32ab, SSVE_UNPRED_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x32ac, CME_DRAM_ACCESS	<ul style="list-style-type: none"> <li>cme_system_dram_mem_hit_ratio</li> </ul>	<ul style="list-style-type: none"> <li>CME_System_Memory_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_System</li> </ul>
0x32ad, CME_L1D_LFB_HIT_RW_FHWPRF	<ul style="list-style-type: none"> <li>cme_l1_prefetcher_accuracy</li> <li>cme_l1_prefetcher_timeliness</li> <li>cme_l1_prefetcher_coverage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Prefetcher_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x32af, SME_INT_OTHER_SPEC	<ul style="list-style-type: none"> <li>za_int_other_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x32b0, SME_FP_OTHER_SPEC	<ul style="list-style-type: none"> <li>za_fp_other_percentage</li> </ul>	<ul style="list-style-type: none"> <li>CME_Operation_Mix</li> </ul>	<ul style="list-style-type: none"> <li>CME_FP_Operation</li> </ul>
0x32b1, CME_L1D_CACHE_REFILL_PRFM	-	-	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x32b2, CME_L1D_CACHE_REFILL_HWPRF	<ul style="list-style-type: none"> <li>cme_l1_prefetcher_accuracy</li> </ul>	<ul style="list-style-type: none"> <li>CME_Prefetcher_Effectiveness</li> </ul>	<ul style="list-style-type: none"> <li>CME_L1D_Cache</li> </ul>
0x32b3, CME_L3D_CACHE_PRFM	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x32b4, CME_L3D_CACHE_HWPRF	-	-	<ul style="list-style-type: none"> <li>CME_L3D_Cache</li> </ul>
0x74, ASE_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8, INST_RETIRED	-	-	<ul style="list-style-type: none"> <li>CME_Retired</li> </ul>
0x8000, SIMD_INST_RETIRED	-	-	<ul style="list-style-type: none"> <li>CME_Retired</li> </ul>
0x8004, SIMD_INST_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8010, FP_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_FP_Operation</li> </ul>
0x8014, FP_HP_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_FP_Operation</li> </ul>
0x8018, FP_SP_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_FP_Operation</li> </ul>
0x801c, FP_DP_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_FP_Operation</li> </ul>
0x8040, INT_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8043, ASE_SVE_INT_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8056, SVE_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8057, ASE_SVE_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8074, SVE_PRED_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8075, SVE_PRED_EMPTY_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8076, SVE_PRED_FULL_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8077, SVE_PRED_PARTIAL_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8078, SVE_UNPRED_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8079, SVE_PRED_NOT_FULL_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8080, SVE_LDST_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8081, SVE_LD_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8082, SVE_ST_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x80c0, FP_SCALE_OPS_SPEC	-	-	• CME_FP_Operation
0x80d0, FP_SCALE2_OPS_SPEC	-	-	• CME_SVE
0x80d2, FP_HP_SCALE2_OPS_SPEC	-	-	• CME_SVE
0x80d3, FP_BF16_SCALE2_OPS_SPEC	-	-	• CME_SVE
0x80d4, FP_SP_SCALE2_OPS_SPEC	-	-	• CME_SVE
0x80d6, FP_DP_SCALE2_OPS_SPEC	-	-	• CME_SVE
0x80e3, ASE_SVE_INT8_SPEC	-	-	• CME_Spec_Operation
0x80e7, ASE_SVE_INT16_SPEC	-	-	• CME_Spec_Operation
0x80eb, ASE_SVE_INT32_SPEC	-	-	• CME_Spec_Operation
0x80ef, ASE_SVE_INT64_SPEC	-	-	• CME_Spec_Operation
0x8352, SME_FP_SPEC	<ul style="list-style-type: none"> <li>• za_fp_op_percentage</li> <li>• za_fp_mopa_percentage</li> <li>• za_fp_fma_percentage</li> <li>• za_fp_dot_percentage</li> <li>• za_fp_addsub_percentage</li> <li>• za_fp_other_percentage</li> </ul>	• CME_Operation_Mix	• CME_Spec_Operation
0x835c, SME_SPEC	-	-	• CME_Spec_Operation
0x835d, SE_SPEC	-	-	• CME_Spec_Operation
0x835e, SME_INST_SPEC	<ul style="list-style-type: none"> <li>• za_op_percentage</li> <li>• za_int_op_percentage</li> <li>• za_fp_op_percentage</li> </ul>	• CME_Operation_Mix	• CME_Spec_Operation
0x8360, SME_INT8_SPEC	-	-	• CME_Spec_Operation
0x8362, SME_FP_BF16_SPEC	-	-	• CME_FP_Operation
0x8364, SME_INT16_SPEC	-	-	• CME_Spec_Operation
0x8366, SME_FP_HP_SPEC	-	-	• CME_Spec_Operation
0x836a, SME_FP_SP_SPEC	-	-	• CME_FP_Operation
0x8370, SME_FP_ADDSUB_SPEC	• za_fp_addsub_percentage	• CME_Operation_Mix	• CME_FP_Operation
0x8372, SME_FP_FMA_SPEC	• za_fp_fma_percentage	• CME_Operation_Mix	• CME_FP_Operation
0x8374, SME_FP_DOT_SPEC	• za_fp_dot_percentage	• CME_Operation_Mix	• CME_FP_Operation
0x8376, SME_FP_MOPA_SPEC	• za_fp_mopa_percentage	• CME_Operation_Mix	• CME_FP_Operation
0x8378, SME_INT_SPEC	<ul style="list-style-type: none"> <li>• za_int_op_percentage</li> <li>• za_int_mopa_percentage</li> <li>• za_int_dot_percentage</li> <li>• za_int_other_percentage</li> </ul>	• CME_Operation_Mix	• CME_Spec_Operation
0x837a, SME_INT_MUL_SPEC	-	-	• CME_Spec_Operation
0x837c, SME_INT_DOT_SPEC	• za_int_dot_percentage	• CME_Operation_Mix	• CME_Spec_Operation
0x837e, SME_INT_MOPA_SPEC	• za_int_mopa_percentage	• CME_Operation_Mix	• CME_Spec_Operation

Code, Mnemonic	Metrics	Metric Groups	Functional Groups
0x8381, SME_PRED2_NOT_FULL_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8384, SME_PRED2_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8385, SME_PRED2_EMPTY_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8386, SME_PRED2_FULL_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8387, SME_PRED2_PARTIAL_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_SVE</li> </ul>
0x8388, SME_LDST_ZAREG_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8389, SME_LD_ZAREG_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x838a, SME_ST_ZAREG_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x838c, SME_LDST_ZTREG_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x838d, SME_LD_ZTREG_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x838e, SME_ST_ZTREG_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8390, SME_LDST_REG_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8391, SME_LD_REG_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8392, SME_ST_REG_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8394, SME_LDST_TILE_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8395, SME_LD_TILE_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x8396, SME_ST_TILE_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>
0x839a, SME_LUT_SPEC	-	-	<ul style="list-style-type: none"> <li>CME_Spec_Operation</li> </ul>

## 5. Metrics by metric group in C1-SME2

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: 56

Metrics for C1-SME2 are grouped into the following metric groups:

- [CME\\_Topdown\\_L1](#), SME2 unit Topdown Level 1 (3 metrics)
- [CME\\_Topdown\\_Frontend](#), SME2 unit Topdown Frontend (2 metrics)
- [CME\\_Topdown\\_Backend](#), SME2 unit Topdown Backend (5 metrics)
- [CME\\_System\\_Memory\\_Effectiveness](#), SME2 unit System Memory Effectiveness (3 metrics)
- [CME\\_General](#), SME2 unit General (1 metrics)
- [CME\\_MPKI](#), SME2 unit Misses Per Kilo Instructions (3 metrics)
- [CME\\_Miss\\_Ratio](#), SME2 unit Miss Ratio (5 metrics)
- [CME\\_L1D\\_Cache\\_Effectiveness](#), SME2 unit L1 Data Cache Effectiveness (3 metrics)
- [CME\\_L3D\\_Cache\\_Effectiveness](#), SME2 unit L3 cluster Cache Effectiveness (3 metrics)
- [CME\\_Bus\\_Effectiveness](#), SME2 unit Bus Effectiveness (1 metrics)
- [CME\\_LL\\_Cache\\_Effectiveness](#), SME2 unit Last Level Cache Effectiveness (3 metrics)
- [CME\\_Operation\\_Mix](#), SME2 unit Operation Mix (20 metrics)
- [CME\\_IQ\\_Effectiveness](#), SME2 unit Issue Queue Effectiveness (3 metrics)
- [CME\\_Port\\_Utilization](#), Execution Unit Effectiveness (6 metrics)
- [CME\\_Prefetcher\\_Effectiveness](#), SME2 unit Prefetcher Effectiveness (3 metrics)

### 5.1 [CME\\_Topdown\\_L1](#) metrics for C1-SME2

SME2 unit Topdown Level 1. This metric group contains the first set of metrics to begin topdown analysis of application performance on an SME2 unit, which provide the percentage distribution of processor pipeline utilization.

Summary of metrics in [CME\\_Topdown\\_L1](#):

- Total metrics: 3

**Table 5-1: CME\_Topdown\_L1 metrics summary**

Metric	Name	Description
<a href="#">cme_backend_bound</a>	SME2 Backend Bound	This metric is the percentage of cycles that were stalled due to resource constraints in the...
<a href="#">cme_frontend_bound</a>	SME2 Frontend Bound	This metric is the percentage of cycles that were stalled due to resource constraints in the...
<a href="#">cme_retiring</a>	SME2 Retiring	This metric is the percentage of cycles that were not stalled for any reason.

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_backend\_bound, SME2 Backend Bound, metric**

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

#### **Units**

This unit is expressed as percent of sme2 unit cycles.

#### **Formula**

$\text{CME\_STALL\_BACKEND} / \text{CME\_CYCLES} * 100$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_CYCLES](#)  
[CME\\_STALL\\_BACKEND](#)

##### **Metric group**

[CME\\_Topdown\\_L1](#)

##### **Methodology**

Stage 1

### **cme\_frontend\_bound, SME2 Frontend Bound, metric**

This metric is the percentage of cycles that were stalled due to resource constraints in the frontend unit of SME2 unit.

#### **Units**

This unit is expressed as percent of sme2 unit cycles.

#### **Formula**

$\text{CME\_STALL\_FRONTEND} / \text{CME\_CYCLES} * 100$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_CYCLES](#)  
[CME\\_STALL\\_FRONTEND](#)

##### **Metric group**

[CME\\_Topdown\\_L1](#)

Methodology

Stage 1

cme\_retiring, SME2 Retiring, metric

This metric is the percentage of cycles that were not stalled for any reason.

Units

This unit is expressed as percent of sme2 unit cycles.

Formula

$$(CME\_CYCLES - CME\_STALL\_FRONTEND - CME\_STALL\_BACKEND) / CME\_CYCLES * 100$$

Related telemetry artifacts

Events

CME\_CYCLES  
CME\_STALL\_BACKEND  
CME\_STALL\_FRONTEND

Metric group

CME\_Topdown\_L1

Methodology

Stage 1

## 5.2 CME\_Topdown\_Frontend metrics for C1-SME2

SME2 unit Topdown Frontend. This metric group contains a set of metrics to analyse a SME2 unit frontend bound code.

Summary of metrics in CME\_Topdown\_Frontend:

- Total metrics: 2

Table 5-2: CME\_Topdown\_Frontend metrics summary

Metric	Name	Description
cme_frontend_cpu_bound	SME2 Frontend CPU Bound	This metric is the percentage of total cycles stalled in the frontend due to frontend core...
cme_frontend_other_bound	SME2 Frontend Other CPU Bound	This metric is the percentage of total cycles stalled in the frontend due to frontend core...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

cme\_frontend\_cpu\_bound, SME2 Frontend CPU 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 no instructions sent by CPU.

### Units

This unit is expressed as percent of sme2 unit cycles.

### Formula

$$\text{CME\_STALL\_FRONTEND\_CPU} / \text{CME\_STALL\_FRONTEND} * 100$$

### Related telemetry artifacts

#### Events

[CME\\_STALL\\_FRONTEND](#)  
[CME\\_STALL\\_FRONTEND\\_CPU](#)

#### Metric group

[CME\\_Topdown\\_Frontend](#)

#### Methodology

Stage 1

### cme\_frontend\_other\_bound, SME2 Frontend Other CPU 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 no instructions sent by another CPU.

### Units

This unit is expressed as percent of sme2 unit cycles.

### Formula

$$\text{CME\_STALL\_FRONTEND\_OTHER\_CPU} / \text{CME\_STALL\_FRONTEND} * 100$$

### Related telemetry artifacts

#### Events

[CME\\_STALL\\_FRONTEND](#)  
[CME\\_STALL\\_FRONTEND\\_OTHER\\_CPU](#)

#### Metric group

[CME\\_Topdown\\_Frontend](#)

#### Methodology

Stage 1

## 5.3 CME\_Topdown\_Backend metrics for C1-SME2

SME2 unit Topdown Backend. This metric group contains a set of metrics to analyze a SME2 unit backend bound code.

Summary of metrics in CME\_Topdown\_Backend:

- Total metrics: 5

**Table 5-3: CME\_Topdown\_Backend metrics summary**

Metric	Name	Description
<a href="#">cme_backend_core_bound</a>	SME2 Backend Core Bound	This metric is the percentage of total cycles stalled in the backend due to DP issue queues not...
<a href="#">cme_backend_mem_bound</a>	SME2 Backend Memory Bound	This metric is the percentage of total cycles stalled in the backend due to Load-Store issue...
<a href="#">cme_backend_mem_cache_bound</a>	SME2 Backend Cache Bound	This metric is the percentage of total cycles stalled in the backend for memory resource...
<a href="#">cme_backend_mem_store_bound</a>	SME2 Backend Store Bound	This metric is the percentage of total cycles stalled in the backend for memory resource...
<a href="#">cme_backend_prefetch_bound</a>	SME2 Backend Prefetch Bound	This metric is the percentage of total cycles stalled in the backend due to hardware prefetcher...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_backend\_core\_bound, SME2 Backend Core Bound, metric**

This metric is the percentage of total cycles stalled in the backend due to DP issue queues not accepting instructions.

#### **Units**

This unit is expressed as percent of sme2 unit cycles.

#### **Formula**

$\text{CME\_STALL\_BACKEND\_CORE} / \text{CME\_STALL\_BACKEND} * 100$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_STALL\\_BACKEND](#)  
[CME\\_STALL\\_BACKEND\\_CORE](#)

##### **Metric group**

[CME\\_Topdown\\_Backend](#)

##### **Methodology**

Stage 1

### **cme\_backend\_mem\_bound, SME2 Backend Memory Bound, metric**

This metric is the percentage of total cycles stalled in the backend due to Load-Store issue queues not accepting instructions.

#### **Units**

This unit is expressed as percent of sme2 unit cycles.

#### **Formula**

$\text{CME\_STALL\_BACKEND\_MEM} / \text{CME\_STALL\_BACKEND} * 100$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_STALL\\_BACKEND](#)

## CME\_STALL\_BACKEND\_MEM

### Metric group

CME\_Topdown\_Backend

### Methodology

Stage 1

## cme\_backend\_mem\_cache\_bound, SME2 Backend Cache Bound, metric

This metric is the percentage of total cycles stalled in the backend for memory resource constraints, with the oldest instruction in at least one of the LS issue queues waiting for cache arbitration.

### Units

This unit is expressed as percent of sme2 unit cycles.

### Formula

$CME\_STALL\_BACKEND\_MEM\_CACHE / CME\_STALL\_BACKEND\_MEM * 100$

### Related telemetry artifacts

#### Events

CME\_STALL\_BACKEND\_MEM

CME\_STALL\_BACKEND\_MEM\_CACHE

### Metric group

CME\_Topdown\_Backend

### Methodology

Stage 1

## cme\_backend\_mem\_store\_bound, SME2 Backend Store Bound, metric

This metric is the percentage of total cycles stalled in the backend for memory resource constraints, with the oldest instruction in at least one of the LS issue queues waiting for store buffer.

### Units

This unit is expressed as percent of sme2 unit cycles.

### Formula

$CME\_STALL\_BACKEND\_MEM\_STORE / CME\_STALL\_BACKEND\_MEM * 100$

### Related telemetry artifacts

#### Events

CME\_STALL\_BACKEND\_MEM

CME\_STALL\_BACKEND\_MEM\_STORE

### Metric group

CME\_Topdown\_Backend

### Methodology

Stage 1

**cme\_backend\_prefetch\_bound, SME2 Backend Prefetch Bound, metric**

This metric is the percentage of total cycles stalled in the backend due to hardware prefetcher issue queues not accepting instructions.

**Units**

This unit is expressed as percent of sme2 unit cycles.

**Formula**

$$\text{CME\_STALL\_BACKEND\_PF} / \text{CME\_STALL\_BACKEND} * 100$$

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

[CME\\_STALL\\_BACKEND](#)  
[CME\\_STALL\\_BACKEND\\_PF](#)

**Metric group**

[CME\\_Topdown\\_Backend](#)

**Methodology**

Stage 1

## 5.4 CME\_System\_Memory\_Effectiveness metrics for C1-SME2

SME2 unit System Memory Effectiveness. This metric group contains a set of metrics to analyze a system-bound code, awaiting data from the system resources internal or external to the SME2 unit.

Summary of metrics in CME\_System\_Memory\_Effectiveness:

- Total metrics: 3

**Table 5-4: CME\_System\_Memory\_Effectiveness metrics summary**

Metric	Name	Description
<a href="#">cme_system_dram_mem_hit_ratio</a>	SME2 System DRAM Hit Ratio	This metric measures the ratio of DRAM hits to the total memory accesses that missed in the...
<a href="#">cme_system_l3d_cache_hit_ratio</a>	SME2 System L3D Cache Hit Ratio	This metric measures the ratio of L3 cache hits to the total memory accesses that missed in the...
<a href="#">cme_system_ll_cache_hit_ratio</a>	SME2 System Last Level Cache Hit Ratio	This metric measures the ratio of Last level cache hits to the total memory accesses that missed...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

**cme\_system\_dram\_mem\_hit\_ratio, SME2 System DRAM Hit Ratio, metric**

This metric measures the ratio of DRAM hits to the total memory accesses that missed in the private L1 cache of SME2 unit. This metric indicates that the workload is memory bound obtaining data from the local DRAM.

### Units

This unit is expressed as per sme2 l1 cache refill.

### Formula

$\text{CME\_DRAM\_ACCESS} / \text{CME\_L1D\_CACHE\_REFILL}$

### Related telemetry artifacts

#### Events

[CME\\_DRAM\\_ACCESS](#)

[CME\\_L1D\\_CACHE\\_REFILL](#)

#### Metric group

[CME\\_System\\_Memory\\_Effectiveness](#)

#### Methodology

Stage 2

### **cme\_system\_l3d\_cache\_hit\_ratio, SME2 System L3D Cache Hit Ratio, metric**

This metric measures the ratio of L3 cache hits to the total memory accesses that missed in the private L1 cache of SME2 unit. This metric indicates that the workload is memory bound obtaining data from L3 cache.

### Units

This unit is expressed as per sme2 l1 cache refill.

### Formula

$\text{CME\_L3D\_CACHE\_HIT} / \text{CME\_L1D\_CACHE\_REFILL}$

### Related telemetry artifacts

#### Events

[CME\\_L1D\\_CACHE\\_REFILL](#)

[CME\\_L3D\\_CACHE\\_HIT](#)

#### Metric group

[CME\\_System\\_Memory\\_Effectiveness](#)

#### Methodology

Stage 2

### **cme\_system\_ll\_cache\_hit\_ratio, SME2 System Last Level Cache Hit Ratio, metric**

This metric measures the ratio of Last level cache hits to the total memory accesses that missed in the private L1 cache of SME2 unit. This metric indicates that the workload is memory bound obtaining data from LL cache.

### Units

This unit is expressed as per sme2 l1 cache refill.

### Formula

$\text{CME\_LL\_CACHE\_HIT} / \text{CME\_L1D\_CACHE\_REFILL}$

Related telemetry artifacts

Events

[CME\\_L1D\\_CACHE\\_REFILL](#)  
[CME\\_LL\\_CACHE\\_HIT](#)

Metric group

[CME\\_System\\_Memory\\_Effectiveness](#)

Methodology

Stage 2

## 5.5 CME\_General metrics for C1-SME2

SME2 unit General. This metric group contains general metrics for SME2 unit performance analysis.

Summary of metrics in CME\_General:

- Total metrics: 1

Table 5-5: CME\_General metrics summary

Metric	Name	Description
<a href="#">cme_ipc</a>	SME2 IPC	This metric measures the number of SME2 instructions retired per SME2 unit cycles.

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

**cme\_ipc, SME2 IPC, metric**

This metric measures the number of SME2 instructions retired per SME2 unit cycles.

Units

This unit is expressed as per sme2 unit cycle.

Formula

[CME\\_INST\\_RETIRED](#) / [CME\\_CYCLES](#)

Related telemetry artifacts

Events

[CME\\_CYCLES](#)  
[CME\\_INST\\_RETIRED](#)

Metric group

[CME\\_General](#)

Methodology

Stage 2

## 5.6 CME\_MPKI metrics for C1-SME2

SME2 unit Misses Per Kilo Instructions. This metric group contains metrics for different SME2 unit resources that can be measured as misses per kilo instructions.

Summary of metrics in CME\_MPKI:

- Total metrics: 3

**Table 5-6: CME\_MPKI metrics summary**

Metric	Name	Description
<a href="#">cme_l1d_cache_mпки</a>	SME2 L1D Cache MPKI	This metric measures the number of SME2 level 1 data cache accesses missed per thousand SME2...
<a href="#">cme_l3d_cache_mпки</a>	SME2 L3D Cache MPKI	This metric measures the number of SME2 level 3 cache accesses missed per thousand SME2...
<a href="#">cme_ll_cache_read_mпки</a>	SME2 LL Cache Read MPKI	This metric measures the number of SME2 last level cache read accesses missed per SME2 thousand...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_l1d\_cache\_mпки, SME2 L1D Cache MPKI, metric**

This metric measures the number of SME2 level 1 data cache accesses missed per thousand SME2 instructions executed.

#### **Units**

This unit is expressed as sme2 mpki.

#### **Formula**

[CME\\_L1D\\_CACHE\\_REFILL](#) / [CME\\_INST\\_RETIRED](#) \* 1000

#### **Related telemetry artifacts**

##### **Events**

[CME\\_INST\\_RETIRED](#)  
[CME\\_L1D\\_CACHE\\_REFILL](#)

##### **Metric group**

[CME\\_MPKI](#)  
Other metric group: [CME\\_L1D\\_Cache\\_Effectiveness](#)

##### **Methodology**

Stage 2

### **cme\_l3d\_cache\_mпки, SME2 L3D Cache MPKI, metric**

This metric measures the number of SME2 level 3 cache accesses missed per thousand SME2 instructions executed.

#### **Units**

This unit is expressed as sme2 mpki.

### Formula

$$\text{CME\_L3D\_CACHE\_REFILL} / \text{CME\_INST\_RETIRED} * 1000$$

### Related telemetry artifacts

#### Events

[CME\\_INST\\_RETIRED](#)  
[CME\\_L3D\\_CACHE\\_REFILL](#)

#### Metric group

[CME\\_MPKI](#)  
Other metric group: [CME\\_L3D\\_Cache\\_Effectiveness](#)

#### Methodology

Stage 2

### cme\_ll\_cache\_read\_mпки, SME2 LL Cache Read MPKI, metric

This metric measures the number of SME2 last level cache read accesses missed per SME2 thousand instructions executed.

### Units

This unit is expressed as sme2 mpki.

### Formula

$$\text{CME\_LL\_CACHE\_MISS\_RD} / \text{CME\_INST\_RETIRED} * 1000$$

### Related telemetry artifacts

#### Events

[CME\\_INST\\_RETIRED](#)  
[CME\\_LL\\_CACHE\\_MISS\\_RD](#)

#### Metric group

[CME\\_MPKI](#)  
Other metric group: [CME\\_LL\\_Cache\\_Effectiveness](#)

#### Methodology

Stage 2

## 5.7 CME\_Miss\_Ratio metrics for C1-SME2

SME2 unit Miss Ratio. This metric group contains metrics to measure miss ratios of different SME2 unit resources.

Summary of metrics in CME\_Miss\_Ratio:

- Total metrics: 5

**Table 5-7: CME\_Miss\_Ratio metrics summary**

Metric	Name	Description
<a href="#">cme_l1d_cache_hit_ratio</a>	SME2 L1D Cache Hit Ratio	This metric measures the ratio of level 1 data cache access hits to the total number of level 1...
<a href="#">cme_l1d_cache_miss_ratio</a>	SME2 L1D Cache Miss Ratio	This metric measures the ratio of SME2 level 1 data cache accesses missed to the total number of...
<a href="#">cme_l3d_cache_hit_ratio</a>	SME2 L3D Cache Hit Ratio	This metric measures the ratio of SME2 level 3 cache access hits to the total number of level 3...
<a href="#">cme_l3d_cache_miss_ratio</a>	SME2 L3D Cache Miss Ratio	This metric measures the ratio of SME2 level 3 cache accesses missed to the total number of level...
<a href="#">cme_ll_cache_read_miss_ratio</a>	LL Cache Read Miss Ratio	This metric measures the ratio of SME2 last level cache read accesses missed to the total number...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_l1d\_cache\_hit\_ratio, SME2 L1D Cache Hit Ratio, metric**

This metric measures the ratio of level 1 data cache access hits 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 sme2 unit cache access.

#### **Formula**

[CME\\_L1D\\_CACHE\\_HIT](#) / [CME\\_L1D\\_CACHE](#)

#### **Related telemetry artifacts**

##### **Events**

[CME\\_L1D\\_CACHE](#)

[CME\\_L1D\\_CACHE\\_HIT](#)

##### **Metric group**

[CME\\_Miss\\_Ratio](#)

Other metric group: [CME\\_L1D\\_Cache\\_Effectiveness](#)

##### **Methodology**

Stage 2

### **cme\_l1d\_cache\_miss\_ratio, SME2 L1D Cache Miss Ratio, metric**

This metric measures the ratio of SME2 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 sme2 unit cache access.

#### **Formula**

[CME\\_L1D\\_CACHE\\_REFILL](#) / [CME\\_L1D\\_CACHE](#)

## Related telemetry artifacts

### Events

[CME\\_L1D\\_CACHE](#)  
[CME\\_L1D\\_CACHE\\_REFILL](#)

### Metric group

[CME\\_Miss\\_Ratio](#)  
Other metric group: [CME\\_L1D\\_Cache\\_Effectiveness](#)

### Methodology

Stage 2

## cme\_l3d\_cache\_hit\_ratio, SME2 L3D Cache Hit Ratio, metric

This metric measures the ratio of SME2 level 3 cache access hits to the total number of level 3 cache accesses, requested by the SME2 unit.

### Units

This unit is expressed as per sme2 unit cache access.

### Formula

$$\frac{\text{CME\_L3D\_CACHE\_HIT}}{\text{CME\_L3D\_CACHE}}$$

## Related telemetry artifacts

### Events

[CME\\_L3D\\_CACHE](#)  
[CME\\_L3D\\_CACHE\\_HIT](#)

### Metric group

[CME\\_Miss\\_Ratio](#)  
Other metric group: [CME\\_L3D\\_Cache\\_Effectiveness](#)

### Methodology

Stage 2

## cme\_l3d\_cache\_miss\_ratio, SME2 L3D Cache Miss Ratio, metric

This metric measures the ratio of SME2 level 3 cache accesses missed to the total number of level 3 cache accesses, requested by the SME2 unit.

### Units

This unit is expressed as per sme2 unit cache access.

### Formula

$$\frac{\text{CME\_L3D\_CACHE\_REFILL}}{\text{CME\_L3D\_CACHE}}$$

## Related telemetry artifacts

### Events

[CME\\_L3D\\_CACHE](#)  
[CME\\_L3D\\_CACHE\\_REFILL](#)

**Metric group**

[CME\\_Miss\\_Ratio](#)

Other metric group: [CME\\_L3D\\_Cache\\_Effectiveness](#)

**Methodology**

Stage 2

**cme\_ll\_cache\_read\_miss\_ratio, LL Cache Read Miss Ratio, metric**

This metric measures the ratio of SME2 last level cache read accesses missed to the total number of last level cache accesses requested by the SME2 unit.

**Units**

This unit is expressed as per sme2 unit cache access.

**Formula**

$$\text{CME\_LL\_CACHE\_MISS\_RD} / \text{CME\_LL\_CACHE\_RD}$$

**Related telemetry artifacts**

**Events**

[CME\\_LL\\_CACHE\\_MISS\\_RD](#)

[CME\\_LL\\_CACHE\\_RD](#)

**Metric group**

[CME\\_Miss\\_Ratio](#)

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

**Methodology**

Stage 2

## 5.8 CME\_L1D\_Cache\_Effectiveness metrics for C1-SME2

SME2 unit L1 Data Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of L1 Data Cache on the SME2 unit.

Summary of metrics in CME\_L1D\_Cache\_Effectiveness:

- Total metrics: 3

**Table 5-8: CME\_L1D\_Cache\_Effectiveness metrics summary**

Metric	Name	Description
<a href="#">cme_l1d_cache_hit_ratio</a>	SME2 L1D Cache Hit Ratio	This metric measures the ratio of level 1 data cache access hits to the total number of level 1...
<a href="#">cme_l1d_cache_miss_ratio</a>	SME2 L1D Cache Miss Ratio	This metric measures the ratio of SME2 level 1 data cache accesses missed to the total number of...
<a href="#">cme_l1d_cache_mпки</a>	SME2 L1D Cache MPKI	This metric measures the number of SME2 level 1 data cache accesses missed per thousand SME2...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_l1d\_cache\_hit\_ratio\*\*, SME2 L1D Cache Hit Ratio, metric**

This metric measures the ratio of level 1 data cache access hits 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 sme2 unit cache access.

#### **Formula**

[CME\\_L1D\\_CACHE\\_HIT](#) / [CME\\_L1D\\_CACHE](#)

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

#### **Related telemetry artifacts**

##### **Events**

[CME\\_L1D\\_CACHE](#)

[CME\\_L1D\\_CACHE\\_HIT](#)

##### **Metric group**

[CME\\_L1D\\_Cache\\_Effectiveness](#)

Other metric group: [CME\\_Miss\\_Ratio](#)

##### **Methodology**

Stage 2

### **cme\_l1d\_cache\_miss\_ratio\*\*, SME2 L1D Cache Miss Ratio, metric**

This metric measures the ratio of SME2 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 sme2 unit cache access.

#### **Formula**

[CME\\_L1D\\_CACHE\\_REFILL](#) / [CME\\_L1D\\_CACHE](#)

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

#### **Related telemetry artifacts**

##### **Events**

[CME\\_L1D\\_CACHE](#)

[CME\\_L1D\\_CACHE\\_REFILL](#)

##### **Metric group**

[CME\\_L1D\\_Cache\\_Effectiveness](#)

Other metric group: [CME\\_Miss\\_Ratio](#)

##### **Methodology**

Stage 2

### **cme\_l1d\_cache\_mpki\*\*, SME2 L1D Cache MPKI, metric**

This metric measures the number of SME2 level 1 data cache accesses missed per thousand SME2 instructions executed.

#### **Units**

This unit is expressed as sme2 mpki.

#### **Formula**

$$\text{CME\_L1D\_CACHE\_REFILL} / \text{CME\_INST\_RETIRED} * 1000$$

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

#### **Related telemetry artifacts**

##### **Events**

[CME\\_INST\\_RETIRED](#)  
[CME\\_L1D\\_CACHE\\_REFILL](#)

##### **Metric group**

[CME\\_L1D\\_Cache\\_Effectiveness](#)  
Other metric group: [CME\\_MPKI](#)

##### **Methodology**

Stage 2

## **5.9 CME\_L3D\_Cache\_Effectiveness metrics for C1-SME2**

SME2 unit L3 cluster Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of L3 cluster cache accesses from SME2 unit on this processor.

Summary of metrics in CME\_L3D\_Cache\_Effectiveness:

- Total metrics: 3

**Table 5-9: CME\_L3D\_Cache\_Effectiveness metrics summary**

Metric	Name	Description
<a href="#">cme_l3d_cache_hit_ratio</a>	SME2 L3D Cache Hit Ratio	This metric measures the ratio of SME2 level 3 cache access hits to the total number of level 3...
<a href="#">cme_l3d_cache_miss_ratio</a>	SME2 L3D Cache Miss Ratio	This metric measures the ratio of SME2 level 3 cache accesses missed to the total number of level...
<a href="#">cme_l3d_cache_mpki</a>	SME2 L3D Cache MPKI	This metric measures the number of SME2 level 3 cache accesses missed per thousand SME2...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_l3d\_cache\_hit\_ratio\*\*, SME2 L3D Cache Hit Ratio, metric**

This metric measures the ratio of SME2 level 3 cache access hits to the total number of level 3 cache accesses, requested by the SME2 unit.

## Units

This unit is expressed as per sme2 unit cache access.

## Formula

$$\text{CME\_L3D\_CACHE\_HIT} / \text{CME\_L3D\_CACHE}$$

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

## Related telemetry artifacts

### Events

[CME\\_L3D\\_CACHE](#)

[CME\\_L3D\\_CACHE\\_HIT](#)

### Metric group

[CME\\_L3D\\_Cache\\_Effectiveness](#)

Other metric group: [CME\\_Miss\\_Ratio](#)

### Methodology

Stage 2

## cme\_l3d\_cache\_miss\_ratio\*\*, SME2 L3D Cache Miss Ratio, metric

This metric measures the ratio of SME2 level 3 cache accesses missed to the total number of level 3 cache accesses, requested by the SME2 unit.

## Units

This unit is expressed as per sme2 unit cache access.

## Formula

$$\text{CME\_L3D\_CACHE\_REFILL} / \text{CME\_L3D\_CACHE}$$

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

## Related telemetry artifacts

### Events

[CME\\_L3D\\_CACHE](#)

[CME\\_L3D\\_CACHE\\_REFILL](#)

### Metric group

[CME\\_L3D\\_Cache\\_Effectiveness](#)

Other metric group: [CME\\_Miss\\_Ratio](#)

### Methodology

Stage 2

## cme\_l3d\_cache\_mpki\*\*, SME2 L3D Cache MPKI, metric

This metric measures the number of SME2 level 3 cache accesses missed per thousand SME2 instructions executed.

## Units

This unit is expressed as sme2 mpki.

**Formula**

$$\text{CME\_L3D\_CACHE\_REFILL} / \text{CME\_INST\_RETIRED} * 1000$$

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

**Related telemetry artifacts**

**Events**

[CME\\_INST\\_RETIRED](#)  
[CME\\_L3D\\_CACHE\\_REFILL](#)

**Metric group**

[CME\\_L3D\\_Cache\\_Effectiveness](#)  
Other metric group: [CME\\_MPKI](#)

**Methodology**

Stage 2

## 5.10 CME\_Bus\_Effectiveness metrics for C1-SME2

SME2 unit Bus Effectiveness. This metric group contains metrics to evaluate the effectiveness of bus accesses from the SME2 unit.

Summary of metrics in CME\_Bus\_Effectiveness:

- Total metrics: 1

**Table 5-10: CME\_Bus\_Effectiveness metrics summary**

Metric	Name	Description
<a href="#">cme_bus_access_average_length</a>	SME2 Bus Access Average Length	This metric measures the average length of bus accesses taken for all the bus requests made by...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

**cme\_bus\_access\_average\_length, SME2 Bus Access Average Length, metric**

This metric measures the average length of bus accesses taken for all the bus requests made by SME2 unit

**Units**

This unit is expressed as sme2 unit bus accesses.

**Formula**

$$\text{CME\_BUS\_ACCESS} / \text{CME\_BUS\_REQ}$$

**Related telemetry artifacts**

**Events**

[CME\\_BUS\\_ACCESS](#)  
[CME\\_BUS\\_REQ](#)

**Metric group**  
[CME\\_Bus\\_Effectiveness](#)

**Methodology**  
Stage 2

## 5.11 CME\_LL\_Cache\_Effectiveness metrics for C1-SME2

SME2 unit Last Level Cache Effectiveness. This metric group contains metrics to evaluate the effectiveness of Last Level Cache accesses from the SME2 unit on this processor.

Summary of metrics in CME\_LL\_Cache\_Effectiveness:

- Total metrics: 3

**Table 5-11: CME\_LL\_Cache\_Effectiveness metrics summary**

Metric	Name	Description
<a href="#">cme_ll_cache_read_hit_ratio</a>	SME2 LL Cache Read Hit Ratio	This metric measures the ratio of SME2 last level cache read accesses hit in the cache to the...
<a href="#">cme_ll_cache_read_miss_ratio</a>	LL Cache Read Miss Ratio	This metric measures the ratio of SME2 last level cache read accesses missed to the total number...
<a href="#">cme_ll_cache_read_mпки</a>	SME2 LL Cache Read MPKI	This metric measures the number of SME2 last level cache read accesses missed per SME2 thousand...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_ll\_cache\_read\_hit\_ratio, SME2 LL Cache Read Hit Ratio, metric**

This metric measures the ratio of SME2 last level cache read accesses hit in the cache to the total number of SME2 last level cache accesses.

**Units**  
This unit is expressed as per sme2 unit cache access.

**Formula**  
$$(CME\_LL\_CACHE\_RD - CME\_LL\_CACHE\_MISS\_RD) / CME\_LL\_CACHE\_RD$$

**Related telemetry artifacts**

**Events**  
[CME\\_LL\\_CACHE\\_MISS\\_RD](#)  
[CME\\_LL\\_CACHE\\_RD](#)

**Metric group**  
[CME\\_LL\\_Cache\\_Effectiveness](#)

**Methodology**  
Stage 2

### **cme\_ll\_cache\_read\_miss\_ratio\*\*, LL Cache Read Miss Ratio, metric**

This metric measures the ratio of SME2 last level cache read accesses missed to the total number of last level cache accesses requested by the SME2 unit.

#### **Units**

This unit is expressed as per sme2 unit cache access.

#### **Formula**

[CME\\_LL\\_CACHE\\_MISS\\_RD](#) / [CME\\_LL\\_CACHE\\_RD](#)

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

#### **Related telemetry artifacts**

##### **Events**

[CME\\_LL\\_CACHE\\_MISS\\_RD](#)

[CME\\_LL\\_CACHE\\_RD](#)

##### **Metric group**

[CME\\_LL\\_Cache\\_Effectiveness](#)

Other metric group: [CME\\_Miss\\_Ratio](#)

##### **Methodology**

Stage 2

### **cme\_ll\_cache\_read\_mpki\*\*, SME2 LL Cache Read MPKI, metric**

This metric measures the number of SME2 last level cache read accesses missed per SME2 thousand instructions executed.

#### **Units**

This unit is expressed as sme2 mpki.

#### **Formula**

[CME\\_LL\\_CACHE\\_MISS\\_RD](#) / [CME\\_INST\\_RETIRED](#) \* 1000

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

#### **Related telemetry artifacts**

##### **Events**

[CME\\_INST\\_RETIRED](#)

[CME\\_LL\\_CACHE\\_MISS\\_RD](#)

##### **Metric group**

[CME\\_LL\\_Cache\\_Effectiveness](#)

Other metric group: [CME\\_MPKI](#)

##### **Methodology**

Stage 2

## 5.12 CME\_Operation\_Mix metrics for C1-SME2

SME2 unit Operation Mix. This metric group provides the distribution of micro-operation types executed by the SME2 unit for the program.

Summary of metrics in CME\_Operation\_Mix:

- Total metrics: 20

**Table 5-12: CME\_Operation\_Mix metrics summary**

Metric	Name	Description
cme_load_percentage	SME2 Load Operations Percentage	This metric measures SME2 load operations as a percentage of SME2 operations speculatively executed.
cme_store_percentage	SME2 Store Operations Percentage	This metric measures store operations as a percentage of operations speculatively executed.
streaming_fp_op_percentage	Streaming FP Operations Percentage	This metric measures streaming floating point operations as a percentage of streaming operations...
streaming_int_op_percentage	Streaming Integer Operations Percentage	This metric measures streaming integer operations as a percentage of streaming operations...
streaming_ld_op_percentage	Streaming Load Operations Percentage	This metric measures streaming load operations as a percentage of streaming operations...
streaming_ls_op_percentage	Streaming Load/Store Operations Percentage	This metric measures streaming load and store operations as a percentage of streaming operations...
streaming_op_percentage	Streaming Operations Percentage	This metric measures streaming operations as a percentage of operations speculatively...
streaming_slow_inst_percentage	Streaming Slow Instructions Percentage	This metric measures streaming slow instructions as a percentage of streaming operations...
streaming_st_op_percentage	Streaming Store Operations Percentage	This metric measures streaming store operations as a percentage of streaming operations...
za_fp_addsub_percentage	Streaming ZA Add Sub Floating Point Operations Percentage	This metric measures SME/SVE classified floating point addition or subtraction operations that...
za_fp_dot_percentage	Streaming ZA Dot Floating Point Operations Percentage	This metric measures SME/SVE classified floating point dot product operations that used ZA arrays...
za_fp_fma_percentage	Streaming ZA FMA Floating Point Operations Percentage	This metric measures SME/SVE classified floating point multiply-add, multiply-add long,...
za_fp_mopa_percentage	Streaming ZA MOPA Floating Point Operations Percentage	This metric measures SME/SVE classified floating point outer product and accumulate, or...
za_fp_op_percentage	Streaming ZA Floating Point Operations Percentage	This metric measures SME/SVE classified floating point operations and used ZA arrays as a...
za_fp_other_percentage	Streaming ZA Other Floating Point Operations Percentage	This metric measures SME/SVE classified floating point operations that used ZA arrays not counted...
za_int_dot_percentage	Streaming ZA Dot Integer Operations Percentage	This metric measures SME integer dot product operations that used ZA arrays as a percentage...
za_int_mopa_percentage	Streaming ZA MOPA Integer Operations Percentage	This metric measures SME integer outer product and accumulate, or outer product and subtract...
za_int_op_percentage	Streaming ZA Integer Operations Percentage	This metric measures SME integer operations that used ZA arrays as a percentage of all the SME...

Metric	Name	Description
<a href="#">za_int_other_percentage</a>	Streaming ZA Other Integer Operations Percentage	This metric measures SME integer operations that used ZA arrays not counted in any other category...
<a href="#">za_op_percentage</a>	Streaming ZA Operations Percentage	This metric measures SME operations that used ZA arrays as a percentage of operations...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_load\_percentage, SME2 Load Operations Percentage, metric**

This metric measures SME2 load operations as a percentage of SME2 operations speculatively executed.

#### **Units**

This unit is expressed as percent of operations.

#### **Formula**

$\text{CME\_LD\_SPEC} / \text{CME\_OP\_RETIRED} * 100$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_LD\\_SPEC](#)

[CME\\_OP\\_RETIRED](#)

##### **Metric group**

[CME\\_Operation\\_Mix](#)

##### **Methodology**

Stage 2

### **cme\_store\_percentage, SME2 Store Operations Percentage, metric**

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

#### **Units**

This unit is expressed as percent of operations.

#### **Formula**

$\text{CME\_ST\_SPEC} / \text{CME\_OP\_RETIRED} * 100$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_OP\\_RETIRED](#)

[CME\\_ST\\_SPEC](#)

##### **Metric group**

[CME\\_Operation\\_Mix](#)

##### **Methodology**

Stage 2

### **streaming\_fp\_op\_percentage, Streaming FP Operations Percentage, metric**

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

#### **Units**

This unit is expressed as percent of streaming operations.

#### **Formula**

$$\text{SSVE\_FP\_SPEC} / \text{SSVE\_INST\_SPEC} * 100$$

#### **Related telemetry artifacts**

##### **Events**

[SSVE\\_FP\\_SPEC](#)  
[SSVE\\_INST\\_SPEC](#)

##### **Metric group**

[CME\\_Operation\\_Mix](#)

##### **Methodology**

Stage 2

### **streaming\_int\_op\_percentage, Streaming Integer Operations Percentage, metric**

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

#### **Units**

This unit is expressed as percent of streaming operations.

#### **Formula**

$$\text{SSVE\_INT\_SPEC} / \text{SSVE\_INST\_SPEC} * 100$$

#### **Related telemetry artifacts**

##### **Events**

[SSVE\\_INST\\_SPEC](#)  
[SSVE\\_INT\\_SPEC](#)

##### **Metric group**

[CME\\_Operation\\_Mix](#)

##### **Methodology**

Stage 2

### **streaming\_ld\_op\_percentage, Streaming Load Operations Percentage, metric**

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

#### **Units**

This unit is expressed as percent of streaming operations.

#### **Formula**

$$\text{SSVE\_LD\_SPEC} / \text{CME\_OP\_RETIRED} * 100$$

## Related telemetry artifacts

### Events

[CME\\_OP\\_RETIRED](#)

[SSVE\\_LD\\_SPEC](#)

### Metric group

[CME\\_Operation\\_Mix](#)

### Methodology

Stage 2

## **streaming\_ls\_op\_percentage, Streaming Load/Store Operations Percentage, metric**

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

### Units

This unit is expressed as percent of streaming operations.

### Formula

$\text{SSVE\_LD\_SPEC} / \text{CME\_OP\_RETIRED} * 100$

## Related telemetry artifacts

### Events

[CME\\_OP\\_RETIRED](#)

[SSVE\\_LDST\\_SPEC](#)

### Metric group

[CME\\_Operation\\_Mix](#)

### Methodology

Stage 2

## **streaming\_op\_percentage, Streaming Operations Percentage, metric**

This metric measures streaming operations as a percentage of operations speculatively executed. Streaming operations include Advanced SIMD, Scalable vector or or Scalable Matrix extension data processing when the CPU is in streaming mode.

### Units

This unit is expressed as percent of operations.

### Formula

$\text{SSVE\_INST\_SPEC} / \text{INST\_SPEC} * 100$

## Related telemetry artifacts

### Events

[INST\\_SPEC](#)

[SSVE\\_INST\\_SPEC](#)

### Metric group

[CME\\_Operation\\_Mix](#)

## Methodology

### Stage 2

#### **streaming\_slow\_inst\_percentage, Streaming Slow Instructions Percentage, metric**

This metric measures streaming slow instructions as a percentage of streaming operations speculatively executed.

#### Units

This unit is expressed as percent of streaming operations.

#### Formula

$$\text{SSVE\_SLOW\_INSTR} / \text{SSVE\_INST\_SPEC} * 100$$

#### Related telemetry artifacts

##### Events

SSVE\_INST\_SPEC  
SSVE\_SLOW\_INSTR

##### Metric group

CME\_Operation\_Mix

##### Methodology

### Stage 2

#### **streaming\_st\_op\_percentage, Streaming Store Operations Percentage, metric**

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

#### Units

This unit is expressed as percent of streaming operations.

#### Formula

$$\text{SSVE\_ST\_SPEC} / \text{CME\_OP\_RETIRED} * 100$$

#### Related telemetry artifacts

##### Events

CME\_OP\_RETIRED  
SSVE\_ST\_SPEC

##### Metric group

CME\_Operation\_Mix

##### Methodology

### Stage 2

#### **za\_fp\_addsub\_percentage, Streaming ZA Add Sub Floating Point Operations Percentage, metric**

This metric measures SME/SVE classified floating point addition or subtraction operations that used ZA arrays as a percentage of all the SME floating point ZA operations speculatively executed.

## Units

This unit is expressed as percent of streaming floating point za operations.

## Formula

$$\text{SME\_FP\_ADDSUB\_SPEC} / \text{SME\_FP\_SPEC} * 100$$

## Related telemetry artifacts

### Events

[SME\\_FP\\_ADDSUB\\_SPEC](#)

[SME\\_FP\\_SPEC](#)

### Metric group

[CME\\_Operation\\_Mix](#)

### Methodology

Stage 2

## za\_fp\_dot\_percentage, Streaming ZA Dot Floating Point Operations Percentage, metric

This metric measures SME/SVE classified floating point dot product operations that used ZA arrays as a percentage of all the SME floating point ZA operations speculatively executed.

## Units

This unit is expressed as percent of streaming floating point za operations.

## Formula

$$\text{SME\_FP\_DOT\_SPEC} / \text{SME\_FP\_SPEC} * 100$$

## Related telemetry artifacts

### Events

[SME\\_FP\\_DOT\\_SPEC](#)

[SME\\_FP\\_SPEC](#)

### Metric group

[CME\\_Operation\\_Mix](#)

### Methodology

Stage 2

## za\_fp\_fma\_percentage, Streaming ZA FMA Floating Point Operations Percentage, metric

This metric measures SME/SVE classified floating point multiply-add, multiply-add long, multiply-subtract and multiply-subtract long operations that used ZA arrays as a percentage of all the SME floating point ZA operations speculatively executed.

## Units

This unit is expressed as percent of streaming floating point za operations.

## Formula

$$\text{SME\_FP\_FMA\_SPEC} / \text{SME\_FP\_SPEC} * 100$$

## Related telemetry artifacts

### Events

[SME\\_FP\\_FMA\\_SPEC](#)

[SME\\_FP\\_SPEC](#)

### Metric group

[CME\\_Operation\\_Mix](#)

### Methodology

Stage 2

## za\_fp\_mopa\_percentage, Streaming ZA MOPA Floating Point Operations Percentage, metric

This metric measures SME/SVE classified floating point outer product and accumulate, or outer product and subtract operations that used ZA arrays as a percentage of all the SME integer ZA operations speculatively executed.

### Units

This unit is expressed as percent of streaming floating point za operations.

### Formula

$\text{SME\_FP\_MOPA\_SPEC} / \text{SME\_FP\_SPEC} * 100$

## Related telemetry artifacts

### Events

[SME\\_FP\\_MOPA\\_SPEC](#)

[SME\\_FP\\_SPEC](#)

### Metric group

[CME\\_Operation\\_Mix](#)

### Methodology

Stage 2

## za\_fp\_op\_percentage, Streaming ZA Floating Point Operations Percentage, metric

This metric measures SME/SVE classified floating point operations and used ZA arrays as a percentage of all the SME operations speculatively executed that used ZA array. ZA arrays are required for instructions that required PSTATE.ZA to be set.

### Units

This unit is expressed as percent of streaming za operations.

### Formula

$\text{SME\_FP\_SPEC} / \text{SME\_INST\_SPEC} * 100$

## Related telemetry artifacts

### Events

[SME\\_FP\\_SPEC](#)

[SME\\_INST\\_SPEC](#)

**Metric group**

CME\_Operation\_Mix

**Methodology**

Stage 2

**za\_fp\_other\_percentage, Streaming ZA Other Floating Point Operations Percentage, metric**

This metric measures SME/SVE classified floating point operations that used ZA arrays not counted in any other category as a percentage of all the SME floating point ZA operations speculatively executed.

**Units**

This unit is expressed as percent of streaming floating point za operations.

**Formula**

$\text{SME\_FP\_OTHER\_SPEC} / \text{SME\_FP\_SPEC} * 100$

**Related telemetry artifacts**

**Events**

SME\_FP\_OTHER\_SPEC

SME\_FP\_SPEC

**Metric group**

CME\_Operation\_Mix

**Methodology**

Stage 2

**za\_int\_dot\_percentage, Streaming ZA Dot Integer Operations Percentage, metric**

This metric measures SME integer dot product operations that used ZA arrays as a percentage of all the SME integer ZA operations speculatively executed.

**Units**

This unit is expressed as percent of streaming integer za operations.

**Formula**

$\text{SME\_INT\_DOT\_SPEC} / \text{SME\_INT\_SPEC} * 100$

**Related telemetry artifacts**

**Events**

SME\_INT\_DOT\_SPEC

SME\_INT\_SPEC

**Metric group**

CME\_Operation\_Mix

**Methodology**

Stage 2

### **za\_int\_mopa\_percentage, Streaming ZA MOPA Integer Operations Percentage, metric**

This metric measures SME integer outer product and accumulate, or outer product and subtract operations that used ZA arrays as a percentage of all the SME integer ZA operations speculatively executed.

#### **Units**

This unit is expressed as percent of streaming integer za operations.

#### **Formula**

$$\text{SME\_INT\_MOPA\_SPEC} / \text{SME\_INT\_SPEC} * 100$$

#### **Related telemetry artifacts**

##### **Events**

[SME\\_INT\\_MOPA\\_SPEC](#)  
[SME\\_INT\\_SPEC](#)

##### **Metric group**

[CME\\_Operation\\_Mix](#)

##### **Methodology**

Stage 2

### **za\_int\_op\_percentage, Streaming ZA Integer Operations Percentage, metric**

This metric measures SME integer operations that used ZA arrays as a percentage of all the SME operations speculatively executed that used ZA array. ZA arrays are required for instructions that required PSTATE.ZA to be set.

#### **Units**

This unit is expressed as percent of streaming sme operations using za tiles.

#### **Formula**

$$\text{SME\_INT\_SPEC} / \text{SME\_INST\_SPEC} * 100$$

#### **Related telemetry artifacts**

##### **Events**

[SME\\_INST\\_SPEC](#)  
[SME\\_INT\\_SPEC](#)

##### **Metric group**

[CME\\_Operation\\_Mix](#)

##### **Methodology**

Stage 2

### **za\_int\_other\_percentage, Streaming ZA Other Integer Operations Percentage, metric**

This metric measures SME integer operations that used ZA arrays not counted in any other category as a percentage of all the SME integer ZA operations speculatively executed.

#### **Units**

This unit is expressed as percent of streaming integer za operations.

Formula

$$\text{SME\_INT\_OTHER\_SPEC} / \text{SME\_INT\_SPEC} * 100$$

Related telemetry artifacts

Events

SME\_INT\_OTHER\_SPEC  
SME\_INT\_SPEC

Metric group

CME\_Operation\_Mix

Methodology

Stage 2

za\_op\_percentage, Streaming ZA Operations Percentage, metric

This metric measures SME operations that used ZA arrays as a percentage of operations speculatively executed. ZA arrays are required for instructions that required PSTATE.ZA to be set.

Units

This unit is expressed as percent of operations.

Formula

$$\text{SME\_INST\_SPEC} / \text{INST\_SPEC} * 100$$

Related telemetry artifacts

Events

INST\_SPEC  
SME\_INST\_SPEC

Metric group

CME\_Operation\_Mix

Methodology

Stage 2

5.13 CME\_IQ\_Effectiveness metrics for C1-SME2

SME2 unit Issue Queue Effectiveness. This metric group provides relative stall cycles by operation type.

Summary of metrics in CME\_IQ\_Effectiveness:

- Total metrics: 3

Table 5-13: CME\_IQ\_Effectiveness metrics summary

Metric	Name	Description
cme_iq_dp0_stall_percentage	IQ DP0 Stall Percentage	This metric measures the percentage of backend stall cycles where at least one operation is...

Metric	Name	Description
<a href="#">cme_iq_dp1_stall_percentage</a>	IQ DP1 Stall Percentage	This metric measures the percentage of backend stall cycles where at least one operation is...
<a href="#">cme_iq_load_stall_percentage</a>	IQ Load Stall Percentage	This metric measures the percentage of backend stall cycles where at least one operation is...

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_iq\_dp0\_stall\_percentage, IQ DP0 Stall Percentage, metric**

This metric measures the percentage of backend stall cycles where at least one operation is waiting to issue to the issue queue DP0, but it is full.

#### **Units**

This unit is expressed as percent of backend stalls.

#### **Formula**

$\text{CME\_DISPATCH\_STALL\_IQ\_DP0} / \text{CME\_STALL\_BACKEND} * 100$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_DISPATCH\\_STALL\\_IQ\\_DP0](#)  
[CME\\_STALL\\_BACKEND](#)

##### **Metric group**

[CME\\_IQ\\_Effectiveness](#)

##### **Methodology**

Stage 2

### **cme\_iq\_dp1\_stall\_percentage, IQ DP1 Stall Percentage, metric**

This metric measures the percentage of backend stall cycles where at least one operation is waiting to issue to the issue queue DP1, but it is full.

#### **Units**

This unit is expressed as percent of backend stalls.

#### **Formula**

$\text{CME\_DISPATCH\_STALL\_IQ\_DP1} / \text{CME\_STALL\_BACKEND} * 100$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_DISPATCH\\_STALL\\_IQ\\_DP1](#)  
[CME\\_STALL\\_BACKEND](#)

##### **Metric group**

[CME\\_IQ\\_Effectiveness](#)

##### **Methodology**

Stage 2

### cme\_iq\_load\_stall\_percentage, IQ Load Stall Percentage, metric

This metric measures the percentage of backend stall cycles where at least one operation is waiting to issue to the issue queue LD, but it is full.

#### Units

This unit is expressed as percent of backend stalls.

#### Formula

$$\text{CME\_DISPATCH\_STALL\_IQ\_LD} / \text{CME\_STALL\_BACKEND} * 100$$

#### Related telemetry artifacts

##### Events

[CME\\_DISPATCH\\_STALL\\_IQ\\_LD](#)  
[CME\\_STALL\\_BACKEND](#)

##### Metric group

[CME\\_IQ\\_Effectiveness](#)

##### Methodology

Stage 2

## 5.14 CME\_Port\_Utilization metrics for C1-SME2

Execution Unit Effectiveness. This metric group provides relative utilization of execution pipes for the program.

Summary of metrics in CME\_Port\_Utilization:

- Total metrics: 6

**Table 5-14: CME\_Port\_Utilization metrics summary**

Metric	Name	Description
<a href="#">cme_alu_port_utilization</a>	SME2 Arithmetic EU Utilization	This metric measures the average number of operations executed by the ALU execution unit per cycle
<a href="#">cme_mac_port_utilization</a>	SME2 MAC EU Utilization	This metric measures the average number of operations executed by the MAC execution unit per cycle
<a href="#">cme_mmdp_port_utilization</a>	Matmul Datapath EU Utilization	This metric measures the average number of operations executed by the Matmul data path execution...
<a href="#">cme_mmmv_port_utilization</a>	Matmul Move EU Utilization	This metric measures the average number of operations executed by the Matmul move execution unit...
<a href="#">cme_perm_port_utilization</a>	Permute EU Utilization	This metric measures the average number of operations executed by the permute execution unit per...
<a href="#">cme_st_port_utilization</a>	Store EU Utilization	This metric measures the average number of operations executed by the store execution unit per cycle

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

### **cme\_alu\_port\_utilization, SME2 Arithmetic EU Utilization, metric**

This metric measures the average number of operations executed by the ALU execution unit per cycle

#### **Units**

This unit is expressed as operations per sme2 unit cycle.

#### **Formula**

$\text{CME\_OP\_ALU\_ISSUE} / \text{CME\_CYCLES}$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_CYCLES](#)

[CME\\_OP\\_ALU\\_ISSUE](#)

##### **Metric group**

[CME\\_Port\\_Utilization](#)

##### **Methodology**

Stage 2

### **cme\_mac\_port\_utilization, SME2 MAC EU Utilization, metric**

This metric measures the average number of operations executed by the MAC execution unit per cycle

#### **Units**

This unit is expressed as operations per sme2 unit cycle.

#### **Formula**

$\text{CME\_OP\_MAC\_ISSUE} / \text{CME\_CYCLES}$

#### **Related telemetry artifacts**

##### **Events**

[CME\\_CYCLES](#)

[CME\\_OP\\_MAC\\_ISSUE](#)

##### **Metric group**

[CME\\_Port\\_Utilization](#)

##### **Methodology**

Stage 2

### **cme\_mmdp\_port\_utilization, Matmul Datapath EU Utilization, metric**

This metric measures the average number of operations executed by the Matmul data path execution unit per cycle

#### **Units**

This unit is expressed as operations per sme2 unit cycle.

#### **Formula**

$\text{CME\_OP\_MMDP\_ISSUE} / \text{CME\_CYCLES}$

## Related telemetry artifacts

### Events

[CME\\_CYCLES](#)  
[CME\\_OP\\_MMDP\\_ISSUE](#)

### Metric group

[CME\\_Port\\_Utilization](#)

### Methodology

Stage 2

## cme\_mmmv\_port\_utilization, Matmul Move EU Utilization, metric

This metric measures the average number of operations executed by the Matmul move execution unit per cycle

### Units

This unit is expressed as operations per sme2 unit cycle.

### Formula

[CME\\_OP\\_MMMV\\_ISSUE](#) / [CME\\_CYCLES](#)

## Related telemetry artifacts

### Events

[CME\\_CYCLES](#)  
[CME\\_OP\\_MMMV\\_ISSUE](#)

### Metric group

[CME\\_Port\\_Utilization](#)

### Methodology

Stage 2

## cme\_perm\_port\_utilization, Permute EU Utilization, metric

This metric measures the average number of operations executed by the permute execution unit per cycle

### Units

This unit is expressed as operations per sme2 unit cycle.

### Formula

[CME\\_OP\\_PERM\\_ISSUE](#) / [CME\\_CYCLES](#)

## Related telemetry artifacts

### Events

[CME\\_CYCLES](#)  
[CME\\_OP\\_PERM\\_ISSUE](#)

### Metric group

[CME\\_Port\\_Utilization](#)

**Methodology**

Stage 2

**cme\_st\_port\_utilization, Store EU Utilization, metric**

This metric measures the average number of operations executed by the store execution unit per cycle

**Units**

This unit is expressed as operations per sme2 unit cycle.

**Formula**

$$\text{CME\_OP\_ST\_ISSUE} / \text{CME\_CYCLES}$$

**Related telemetry artifacts**

**Events**

[CME\\_CYCLES](#)  
[CME\\_OP\\_ST\\_ISSUE](#)

**Metric group**

[CME\\_Port\\_Utilization](#)

**Methodology**

Stage 2

## 5.15 CME\_Prefetcher\_Effectiveness metrics for C1-SME2

SME2 unit Prefetcher Effectiveness. This metric group provides the SME2 unit prefetcher effectiveness metrics.

Summary of metrics in CME\_Prefetcher\_Effectiveness:

- Total metrics: 3

**Table 5-15: CME\_Prefetcher\_Effectiveness metrics summary**

Metric	Name	Description
<a href="#">cme_l1_prefetcher_accuracy</a>	SME2 L1D Cache Prefetcher Accuracy	This metric measures SME2 L1D cache prefetcher accuracy
<a href="#">cme_l1_prefetcher_coverage</a>	SME2 L1D Cache Prefetcher Coverage	This metric measures L1D cache prefetcher coverage
<a href="#">cme_l1_prefetcher_timeliness</a>	SME2 L1D Cache Prefetcher Timeliness	This metric measures SME2 L1D cache prefetcher timeliness

For a complete list of the metrics in C1-SME2, see [Metrics cheat sheet for C1-SME2](#) and [Metrics lookup table for C1-SME2](#).

**cme\_l1\_prefetcher\_accuracy, SME2 L1D Cache Prefetcher Accuracy, metric**

This metric measures SME2 L1D cache prefetcher accuracy

**Units**

This unit is expressed as useful prefetches per total prefetches.

### Formula

$$\frac{\text{CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF} + \text{CME\_L1D\_LFB\_HIT\_RW\_FWWPRF}}{\text{CME\_L1D\_CACHE\_REFILL\_HWPRF}}$$

### Related telemetry artifacts

#### Events

[CME\\_L1D\\_CACHE\\_HIT\\_RW\\_FWWPRF](#)  
[CME\\_L1D\\_CACHE\\_REFILL\\_HWPRF](#)  
[CME\\_L1D\\_LFB\\_HIT\\_RW\\_FWWPRF](#)

#### Metric group

[CME\\_Prefetcher\\_Effectiveness](#)

#### Methodology

Stage 2

### **cme\_l1\_prefetcher\_coverage, SME2 L1D Cache Prefetcher Coverage, metric**

This metric measures L1D cache prefetcher coverage

### Units

This unit is expressed as useful prefetches per demand misses.

### Formula

$$\frac{\text{CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF} + \text{CME\_L1D\_LFB\_HIT\_RW\_FWWPRF}}{(\text{CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF} + \text{CME\_L1D\_LFB\_HIT\_RW\_FWWPRF} + \text{CME\_L1D\_CACHE\_REFILL\_RD} + \text{CME\_L1D\_CACHE\_REFILL\_WR})}$$

### Related telemetry artifacts

#### Events

[CME\\_L1D\\_CACHE\\_HIT\\_RW\\_FWWPRF](#)  
[CME\\_L1D\\_CACHE\\_REFILL\\_RD](#)  
[CME\\_L1D\\_CACHE\\_REFILL\\_WR](#)  
[CME\\_L1D\\_LFB\\_HIT\\_RW\\_FWWPRF](#)

#### Metric group

[CME\\_Prefetcher\\_Effectiveness](#)

#### Methodology

Stage 2

### **cme\_l1\_prefetcher\_timeliness, SME2 L1D Cache Prefetcher Timeliness, metric**

This metric measures SME2 L1D cache prefetcher timeliness

### Units

This unit is expressed as timely prefetches per useful prefetches.

### Formula

$$\frac{\text{CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF}}{(\text{CME\_L1D\_CACHE\_HIT\_RW\_FWWPRF} + \text{CME\_L1D\_LFB\_HIT\_RW\_FWWPRF})}$$

## Related telemetry artifacts

### Events

[CME\\_L1D\\_CACHE\\_HIT\\_RW\\_FHWPRF](#)

[CME\\_L1D\\_LFB\\_HIT\\_RW\\_FHWPRF](#)

### Metric group

[CME\\_Prefetcher\\_Effectiveness](#)

### Methodology

Stage 2

## 6. PMU events by functional group in C1-SME2

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-SME2 provides the following types of PMU events:

- Total implemented Common events: 67
- Total Implemented Product ImpDef events: 145
- PMU Only events : 212
- ETE Only events : 0

PMU events for C1-SME2 are grouped into the following functional groups:

- [CME\\_Bus](#), SME2 BUS (8 events)
- [CME\\_L1D\\_Cache](#), SME2 L1D CACHE (26 events)
- [CME\\_L3D\\_Cache](#), SME2 L3D CACHE (13 events)
- [CME\\_LL\\_Cache](#), SME2 LL CACHE (4 events)
- [CME\\_Memory](#), SME2 MEMORY (9 events)
- [CME\\_Retired](#), SME2 RETIRED (4 events)
- [CME\\_Spec\\_Operation](#), SME2 SPEC OPERATION (67 events)
- [CME\\_Stall](#), SME2 STALL (12 events)
- [CME\\_General](#), SME2 GENERAL (5 events)
- [CME\\_SVE](#), SME2 SVE (40 events)
- [CME\\_FP\\_Operation](#), CME\_FP\_Operation (12 events)
- [CME\\_IQ\\_Efficiency](#), SME2 ISSUE QUEUE (3 events)
- [CME\\_Port\\_Utilization](#), SME2 PORT UTILIZATION (7 events)
- [CME\\_Coherency](#), SME2 COHERENCY (1 events)
- [CME\\_System](#), SME2 SYSTEM (1 events)

### 6.1 [CME\\_Bus](#) (SME2 BUS) events for C1-SME2

SME2 unit Bus transaction related events.

Summary of events in [CME\\_Bus](#):

- Total implemented Common events: 0

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

**Table 6-1: CME\_Bus events summary**

Code	Mnemonic	Name	Description
0x3266	<a href="#">CME_BUS_REQ_RD_PERCYC</a>	SME2 unit Bus read transactions in progress	Counts memory transaction requests issued by the CPU to the external bus, including...
0x3267	<a href="#">CME_BUS_REQ</a>	SME2 unit Bus request	Counts memory transaction requests issued by the CPU to the external bus, including...
0x3268	<a href="#">CME_BUS_REQ_RD</a>	SME2 unit Bus request, read	Counts memory read transaction requests issued by the CPU to the external bus, including...
0x3269	<a href="#">CME_BUS_REQ_WR</a>	SME2 unit Bus request, write	Counts memory write transaction requests issued by the CPU to the external bus, including...
0x326a	<a href="#">CME_BUS_CYCLES</a>	SME2 unit Bus cycle	Counts bus cycles in the SME2 unit. Bus cycles represent a clock cycle in which a transaction...
0x326b	<a href="#">CME_BUS_ACCESS</a>	SME2 unit Bus access	Counts memory transactions issued by the SME2 unit to the external bus, including snoop requests...
0x326c	<a href="#">CME_BUS_ACCESS_RD</a>	SME2 unit Bus access, read	Counts memory read transactions seen on the external bus. Each beat of data is counted individually.
0x326d	<a href="#">CME_BUS_ACCESS_WR</a>	SME2 unit Bus access, write	Counts memory write transactions seen on the external bus. Each beat of data is counted...

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### **0x3266 CME\_BUS\_REQ\_RD\_PERCYC, SME2 unit Bus read transactions in progress, event**

Counts memory transaction requests issued by the CPU to the external bus, including snoop requests and snoop responses in progress on a processor cycle.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Bus](#)

### **0x3267 CME\_BUS\_REQ, SME2 unit Bus request, event**

Counts memory transaction requests issued by the CPU to the external bus, including snoop requests and snoop responses.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_bus\\_access\\_average\\_length](#)

##### **Metric groups**

[CME\\_Bus\\_Effectiveness](#)

## Functional groups

[CME\\_Bus](#)

### 0x3268 CME\_BUS\_REQ\_RD, SME2 unit Bus request, read, event

Counts memory read transaction requests issued by the CPU to the external bus, including snoop requests and snoop responses.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

## Functional groups

[CME\\_Bus](#)

### 0x3269 CME\_BUS\_REQ\_WR, SME2 unit Bus request, write, event

Counts memory write transaction requests issued by the CPU to the external bus, including snoop requests and snoop responses.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

## Functional groups

[CME\\_Bus](#)

### 0x326a CME\_BUS\_CYCLES, SME2 unit Bus cycle, event

Counts bus cycles in the SME2 unit. Bus cycles represent a clock cycle in which a transaction could be sent or received on the interface from the SME2 unit to the external bus.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

## Functional groups

[CME\\_Bus](#)

### 0x326b CME\_BUS\_ACCESS, SME2 unit Bus access, event

Counts memory transactions issued by the SME2 unit to the external bus, including snoop requests and snoop responses. Each beat of data is counted individually.

#### Related telemetry artifacts

##### Metrics

- [cme\\_bus\\_access\\_average\\_length](#)

##### Metric groups

[CME\\_Bus\\_Effectiveness](#)

## Functional groups

[CME\\_Bus](#)

### 0x326c CME\_BUS\_ACCESS\_RD, SME2 unit Bus access, read, event

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

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Bus](#)

### 0x326d CME\_BUS\_ACCESS\_WR, SME2 unit Bus access, write, event

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

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Bus](#)

## 6.2 CME\_L1D\_Cache (SME2 L1D CACHE) events for C1-SME2

SME2 unit L1 data cache related events.

Summary of events in CME\_L1D\_Cache:

- Total implemented Common events: 0
- Total Implemented Product ImpDef events: 26
- PMU Only events : 26
- ETE Only events : 0

**Table 6-2: CME\_L1D\_Cache events summary**

Code	Mnemonic	Name	Description
0x326f	<a href="#">CME_L1D_CACHE</a>	SME2 unit Level 1 data cache access	Counts each memory-read operation or memory-write operation that causes a level 1 data cache...
0x3270	<a href="#">CME_L1D_CACHE_HIT</a>	Level 1 data cache hit, any access	Counts each access counted by L1D_CACHE that hits in the level 1 data cache.
0x3271	<a href="#">CME_L1D_CACHE_HIT_RW</a>	SME2 unit Level 1 data cache demand hit	Counts cache line hits in the level 1 data cache due to Load or Store operations
0x3272	<a href="#">CME_L1D_CACHE_HIT_RD</a>	SME2 unit Level 1 data cache demand hit, read	Counts cache line hits in the level 1 data cache due to Load operations

Code	Mnemonic	Name	Description
0x3273	CME_L1D_CACHE_HIT_WR	SME2 unit Level 1 data cache demand hit, write	Counts cache line hits in the level 1 data cache due to Store operations
0x3274	CME_L1D_CACHE_HIT_RW_FHWPRF	SME2 unit Level 1 data cache demand access first hit, fetched by hardware prefetcher	The counter counts each demand access first hit counted by L1D_CACHE_HIT_RW where the cache line...
0x3275	CME_L1D_CACHE_MISS	SME2 unit Level 1 data cache demand access miss	Counts access counted by L1D_CACHE that miss in the level 1 data cache.
0x3276	CME_L1D_CACHE_RW	SME2 unit Level 1 data cache demand access	Counts level 1 data demand cache accesses from any load or store operation.
0x3277	CME_L1D_CACHE_RD	SME2 unit Level 1 data cache demand access, read	Counts level 1 data cache accesses from any load operation.
0x3278	CME_L1D_CACHE_WR	SME2 unit Level 1 data cache demand access, write	Counts level 1 data cache accesses generated by store operations.
0x3279	CME_L1D_CACHE_PRF	SME2 unit Level 1 data cache, preload or prefetch hit	Counts level 1 data cache accesses from hardware prefetcher, software preload or prefetch ...
0x327a	CME_L1D_CACHE_HWPRF	SME2 unit Level 1 data cache hardware prefetch	The counter counts each access counted by L1D_CACHE that is due to a hardware prefetch. The...
0x327b	CME_L1D_CACHE_PRFM	SME2 unit Level 1 instruction cache software preload	Counts level 1 data cache accesses from software preload or prefetch instructions.
0x327c	CME_L1D_CACHE_REFILL	SME2 unit Level 1 data cache refill	Counts each access counted by L1D_CACHE that causes a level 1 data cache refill due to a miss in...
0x327d	CME_L1D_CACHE_REFILL_INNER	SME2 unit Level 1 data cache refill, inner	Counts level 1 data cache refills where the cache line data came from caches inside the immediate...
0x327e	CME_L1D_CACHE_REFILL_OUTER	SME2 unit Level 1 data cache refill, outer	Counts level 1 data cache refills for which the cache line data came from outside the immediate...
0x3280	CME_L1D_CACHE_REFILL_RD	SME2 unit Level 1 data cache refill, read	Counts level 1 data cache refills caused by speculatively executed load instructions where the...
0x3281	CME_L1D_CACHE_REFILL_WR	SME2 unit Level 1 data cache refill, write	Counts level 1 data cache refills caused by speculatively executed store instructions where the...
0x3282	CME_L1D_CACHE_INVAL	SME2 unit Level 1 data cache invalidate	Counts each explicit invalidation of a cache line in the level 1 data cache caused by broadcast...
0x3283	CME_L1D_CACHE_LMISS_RD	SME2 unit Level 1 data cache long-latency read miss	Counts cache line refills into the level 1 data cache from any memory read operations, that...
0x3284	CME_L1D_CACHE_WB	SME2 unit Level 1 data cache write-back	Counts write-backs of dirty data from the L1 data cache to the next cache. This occurs when...
0x3285	CME_L1D_CACHE_WB_CLEAN	SME2 unit Level 1 data cache write-back, cleaning and coherency	Counts write-backs from the level 1 data cache that are a result of a coherency operation made by...

Code	Mnemonic	Name	Description
0x3286	<a href="#">CME_L1D_CACHE_WB_VICTIM</a>	SME2 unit Level 1 data cache write-back, victim	Counts dirty cache line evictions from the level 1 data cache caused by a new cache line...
0x32ad	<a href="#">CME_L1D_LFB_HIT_RW_FHWPRF</a>	Level 1 data cache demand access line-fill buffer first hit, recently fetched by hardware prefetcher	The counter counts each demand access first hit in outstanding where the cache line was fetched...
0x32b1	<a href="#">CME_L1D_CACHE_REFILL_PRFM</a>	SME2 unit Level 1 data cache refill, software preload	Counts level 1 data cache refills where the cache line access was generated by software preload...
0x32b2	<a href="#">CME_L1D_CACHE_REFILL_HWPRF</a>	SME2 unit Level 1 data cache refill, hardware preload	Counts each hardware prefetch counted by L1D_CACHE_HWPRF that causes a refill of the Level 1 data...

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### 0x326f CME\_L1D\_CACHE, SME2 unit Level 1 data cache access, event

Counts each memory-read operation or memory-write operation that causes a level 1 data cache accesses.

Each access to a cache line is counted including the multiple accesses caused by single instructions such as multiple-vector instructions. Each access to other level 1 data or unified memory structures, for example refill buffers, write buffers, and write-back buffers, are also counted.

#### Related telemetry artifacts

##### Metrics

- [cme\\_l1d\\_cache\\_miss\\_ratio](#) in [CME\\_L1D\\_Cache\\_Effectiveness](#)
- [cme\\_l1d\\_cache\\_miss\\_ratio](#) in [CME\\_Miss\\_Ratio](#)
- [cme\\_l1d\\_cache\\_hit\\_ratio](#) in [CME\\_L1D\\_Cache\\_Effectiveness](#)
- [cme\\_l1d\\_cache\\_hit\\_ratio](#) in [CME\\_Miss\\_Ratio](#)

##### Metric groups

[CME\\_L1D\\_Cache\\_Effectiveness](#)  
[CME\\_Miss\\_Ratio](#)

##### Functional groups

[CME\\_L1D\\_Cache](#)

### 0x3270 CME\_L1D\_CACHE\_HIT, Level 1 data cache hit, any access, event

Counts each access counted by L1D\_CACHE that hits in the level 1 data cache.

#### Related telemetry artifacts

##### Metrics

- [cme\\_l1d\\_cache\\_hit\\_ratio](#) in [CME\\_L1D\\_Cache\\_Effectiveness](#)
- [cme\\_l1d\\_cache\\_hit\\_ratio](#) in [CME\\_Miss\\_Ratio](#)

### Metric groups

[CME\\_L1D\\_Cache\\_Effectiveness](#)

[CME\\_Miss\\_Ratio](#)

### Functional groups

[CME\\_L1D\\_Cache](#)

## **0x3271 CME\_L1D\_CACHE\_HIT\_RW, SME2 unit Level 1 data cache demand hit, event**

Counts cache line hits in the level 1 data cache due to Load or Store operations

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L1D\\_Cache](#)

## **0x3272 CME\_L1D\_CACHE\_HIT\_RD, SME2 unit Level 1 data cache demand hit, read, event**

Counts cache line hits in the level 1 data cache due to Load operations

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L1D\\_Cache](#)

## **0x3273 CME\_L1D\_CACHE\_HIT\_WR, SME2 unit Level 1 data cache demand hit, write, event**

Counts cache line hits in the level 1 data cache due to Store operations

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L1D\\_Cache](#)

## **0x3274 CME\_L1D\_CACHE\_HIT\_RW\_FHWPRF, SME2 unit Level 1 data cache demand access first hit, fetched by hardware prefetcher, event**

The counter counts each demand access first hit counted by L1D\_CACHE\_HIT\_RW where the cache line was fetched by a hardware prefetcher. That is, the L1D\_CACHE\_REFILL\_HWPRF event was generated when the cache line was fetched into the cache. Only the first hit by a demand access is counted. After this event is generated for a cache line, the event is not generated again for the same cache line while it remains in the cache.

## Related telemetry artifacts

### Metrics

- [cme\\_l1\\_prefetcher\\_accuracy](#)
- [cme\\_l1\\_prefetcher\\_timeliness](#)
- [cme\\_l1\\_prefetcher\\_coverage](#)

### Metric groups

[CME\\_Prefetcher\\_Effectiveness](#)

### Functional groups

[CME\\_L1D\\_Cache](#)

## 0x3275 CME\_L1D\_CACHE\_MISS, SME2 unit Level 1 data cache demand access miss, event

Counts access counted by L1D\_CACHE that miss in the level 1 data cache.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L1D\\_Cache](#)

## 0x3276 CME\_L1D\_CACHE\_RW, SME2 unit Level 1 data cache demand access, event

Counts level 1 data demand cache accesses from any load or store operation.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L1D\\_Cache](#)

## 0x3277 CME\_L1D\_CACHE\_RD, SME2 unit Level 1 data cache demand access, read, event

Counts level 1 data cache accesses from any load operation.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L1D\\_Cache](#)

## 0x3278 CME\_L1D\_CACHE\_WR, SME2 unit Level 1 data cache demand access, write, event

Counts level 1 data cache accesses generated by store operations.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_L1D\\_Cache](#)

### 0x3279 CME\_L1D\_CACHE\_PRF, SME2 unit Level 1 data cache, preload or prefetch hit, event

Counts level 1 data cache accesses from hardware prefetcher, software preload or prefetch instructions.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_L1D\\_Cache](#)

### 0x327a CME\_L1D\_CACHE\_HWPRF, SME2 unit Level 1 data cache hardware prefetch, event

The counter counts each access counted by L1D\_CACHE that is due to a hardware prefetch. The hardware prefetch is generated by a hardware prefetcher at the Level 1 data or unified cache.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_L1D\\_Cache](#)

### 0x327b CME\_L1D\_CACHE\_PRFM, SME2 unit Level 1 instruction cache software preload, event

Counts level 1 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 the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_L1D\\_Cache](#)

### 0x327c CME\_L1D\_CACHE\_REFILL, SME2 unit Level 1 data cache refill, event

Counts each access counted by L1D\_CACHE that causes a level 1 data cache refill due to a miss in the level 1 data cache. This event only counts one event per cache line.

### Related telemetry artifacts

#### Metrics

- [cme\\_l1d\\_cache\\_mpki](#) in [CME\\_L1D\\_Cache\\_Effectiveness](#)

- [cme\\_l1d\\_cache\\_mпки](#) in [CME\\_MPKI](#)
- [cme\\_l1d\\_cache\\_miss\\_ratio](#) in [CME\\_L1D\\_Cache\\_Effectiveness](#)
- [cme\\_l1d\\_cache\\_miss\\_ratio](#) in [CME\\_Miss\\_Ratio](#)
- [cme\\_system\\_dram\\_mem\\_hit\\_ratio](#)
- [cme\\_system\\_l3d\\_cache\\_hit\\_ratio](#)
- [cme\\_system\\_ll\\_cache\\_hit\\_ratio](#)

#### Metric groups

[CME\\_L1D\\_Cache\\_Effectiveness](#)  
[CME\\_MPKI](#)  
[CME\\_Miss\\_Ratio](#)  
[CME\\_System\\_Memory\\_Effectiveness](#)

#### Functional groups

[CME\\_L1D\\_Cache](#)

### **0x327d CME\_L1D\_CACHE\_REFILL\_INNER, SME2 unit Level 1 data cache refill, inner, event**

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

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_L1D\\_Cache](#)

### **0x327e CME\_L1D\_CACHE\_REFILL\_OUTER, SME2 unit Level 1 data cache refill, outer, event**

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

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_L1D\\_Cache](#)

### **0x3280 CME\_L1D\_CACHE\_REFILL\_RD, SME2 unit Level 1 data cache refill, read, event**

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

## Related telemetry artifacts

### Metrics

- [cme\\_l1\\_prefetcher\\_coverage](#)

### Metric groups

[CME\\_Prefetcher\\_Effectiveness](#)

### Functional groups

[CME\\_L1D\\_Cache](#)

## 0x3281 CME\_L1D\_CACHE\_REFILL\_WR, SME2 unit Level 1 data cache refill, write, event

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

## Related telemetry artifacts

### Metrics

- [cme\\_l1\\_prefetcher\\_coverage](#)

### Metric groups

[CME\\_Prefetcher\\_Effectiveness](#)

### Functional groups

[CME\\_L1D\\_Cache](#)

## 0x3282 CME\_L1D\_CACHE\_INVALID, SME2 unit Level 1 data cache invalidate, event

Counts each explicit invalidation of a cache line in the level 1 data cache caused by broadcast cache coherency operations from another CPU in the system.

This event does not count for the following conditions:

1. A cache refill invalidates a cache line.
2. Invalidation during hardware power-off sequence.

## Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L1D\\_Cache](#)

## 0x3283 CME\_L1D\_CACHE\_LMISS\_RD, SME2 unit Level 1 data cache long-latency read miss, event

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

## Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

## Functional groups

[CME\\_L1D\\_Cache](#)

### **0x3284 CME\_L1D\_CACHE\_WB, SME2 unit Level 1 data cache write-back, event**

Counts write-backs of dirty data from the L1 data cache to the next cache. This occurs when either a dirty cache line is evicted from L1 data cache and allocated in the next cache or dirty data is written to the next cache 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 next level of cache without writing L1, such as write streaming mode.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

## Functional groups

[CME\\_L1D\\_Cache](#)

### **0x3285 CME\_L1D\_CACHE\_WB\_CLEAN, SME2 unit Level 1 data cache write-back, cleaning and coherency, event**

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

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

## Functional groups

[CME\\_L1D\\_Cache](#)

### **0x3286 CME\_L1D\_CACHE\_WB\_VICTIM, SME2 unit Level 1 data cache write-back, victim, event**

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

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

## Functional groups

[CME\\_L1D\\_Cache](#)

### **0x32ad CME\_L1D\_LFB\_HIT\_RW\_FHWPRF, Level 1 data cache demand access line-fill buffer first hit, recently fetched by hardware prefetcher, event**

The counter counts each demand access first hit in outstanding where the cache line was fetched by a hardware prefetcher. Only the first hit by a demand access is counted. After this event is generated for a cache line, the event is not generated again for the same cache line while it remains in the cache.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_l1\\_prefetcher\\_accuracy](#)
- [cme\\_l1\\_prefetcher\\_timeliness](#)
- [cme\\_l1\\_prefetcher\\_coverage](#)

##### **Metric groups**

[CME\\_Prefetcher\\_Effectiveness](#)

##### **Functional groups**

[CME\\_L1D\\_Cache](#)

### **0x32b1 CME\_L1D\_CACHE\_REFILL\_PRFM, SME2 unit Level 1 data cache refill, software preload, event**

Counts level 1 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 the C1-SME2 Telemetry Specification.

##### **Functional groups**

[CME\\_L1D\\_Cache](#)

### **0x32b2 CME\_L1D\_CACHE\_REFILL\_HWPRF, SME2 unit Level 1 data cache refill, hardware preload, event**

Counts each hardware prefetch counted by L1D\_CACHE\_HWPRF that causes a refill of the Level 1 data or unified cache from outside of the Level 1 data or unified cache.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_l1\\_prefetcher\\_accuracy](#)

##### **Metric groups**

[CME\\_Prefetcher\\_Effectiveness](#)

##### **Functional groups**

[CME\\_L1D\\_Cache](#)

## 6.3 CME\_L3D\_Cache (SME2 L3D CACHE) events for C1-SME2

SME2 unit L3 data cache related events.

Summary of events in CME\_L3D\_Cache:

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

**Table 6-3: CME\_L3D\_Cache events summary**

Code	Mnemonic	Name	Description
0x3287	CME_L3D_CACHE	Level 3 data cache access due to the SME2 unit	Counts level 3 cache accesses. Level 3 cache is a unified cache for data and instruction...
0x3288	CME_L3D_CACHE_RW	Level 3 data cache demand access due to the SME2 unit	Counts level 3 cache accesses caused by any memory read or write operation. level 3 cache is a...
0x3289	CME_L3D_CACHE_RD	Level 3 data cache demand access due to the SME2 unit, read	Sum of L3 cache reads from all Pipelines
0x328a	CME_L3D_CACHE_WR	Level 3 data cache demand access due to the SME2 unit, write	L3 cache write
0x328b	CME_L3D_CACHE_ALLOCATE	Level 3 data cache allocation due to the SME2 unit, without refill	Counts level 3 cache line allocates that do not fetch data from outside the level 3 data or...
0x328d	CME_L3D_CACHE_LMISS_RD	Level 3 data cache long-latency read miss due to the SME2 unit	Counts any cache line refill into the level 3 cache from memory read operations that incurred...
0x328e	CME_L3D_CACHE_HIT	Level 3 data cache hit due to the SME2 unit	Counts level 3 cache accesses that hit in the level 3 cache.
0x328f	CME_L3D_CACHE_MISS	Level 3 data cache demand access miss due to the SME2 unit	Counts level 3 cache accesses that missed in the level 3 cache.
0x3290	CME_L3D_CACHE_REFILL	Level 3 data cache refill due to the SME2 unit	Counts level 3 accesses that receive data from outside the L3 cache.
0x3291	CME_L3D_CACHE_REFILL_PRFM	Level 3 data cache refill due to the SME2 unit, software preload	Counts cacheable reads generated by hardware or software prefetches that receive data from...
0x3292	CME_L3D_CACHE_REFILL_RD	Level 3 data cache refill due to the SME2 unit, read	<ul style="list-style-type: none"> <li>• This event duplicates L3D_CACHE_REFILL.</li> <li>• If either the core is configured without a per-core...</li> </ul>
0x32b3	CME_L3D_CACHE_PRFM	Level 3 data cache software preload due to the SME2 unit	Counts level 3 cache accesses generated by software prefetches.
0x32b4	CME_L3D_CACHE_HWPRF	SME2 unit Level 3 data cache hardware prefetch	Counts level 3 cache accesses generated by hardware prefetches.

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### **0x3287 CME\_L3D\_CACHE, Level 3 data cache access due to the SME2 unit, 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**

##### **Metrics**

- [cme\\_l3d\\_cache\\_miss\\_ratio](#) in [CME\\_L3D\\_Cache\\_Effectiveness](#)
- [cme\\_l3d\\_cache\\_miss\\_ratio](#) in [CME\\_Miss\\_Ratio](#)
- [cme\\_l3d\\_cache\\_hit\\_ratio](#) in [CME\\_L3D\\_Cache\\_Effectiveness](#)
- [cme\\_l3d\\_cache\\_hit\\_ratio](#) in [CME\\_Miss\\_Ratio](#)

##### **Metric groups**

[CME\\_L3D\\_Cache\\_Effectiveness](#)  
[CME\\_Miss\\_Ratio](#)

##### **Functional groups**

[CME\\_L3D\\_Cache](#)

### **0x3288 CME\_L3D\_CACHE\_RW, Level 3 data cache demand access due to the SME2 unit, event**

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

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

##### **Functional groups**

[CME\\_L3D\\_Cache](#)

### **0x3289 CME\_L3D\_CACHE\_RD, Level 3 data cache demand access due to the SME2 unit, read, event**

Sum of L3 cache reads from all Pipelines

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

##### **Functional groups**

[CME\\_L3D\\_Cache](#)

### **0x328a CME\_L3D\_CACHE\_WR, Level 3 data cache demand access due to the SME2 unit, write, event**

L3 cache write

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_L3D\\_Cache](#)

### 0x328b CME\_L3D\_CACHE\_ALLOCATE, Level 3 data cache allocation due to the SME2 unit, without refill, event

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

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_L3D\\_Cache](#)

### 0x328d CME\_L3D\_CACHE\_LMISS\_RD, Level 3 data cache long-latency read miss due to the SME2 unit, event

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

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_L3D\\_Cache](#)

### 0x328e CME\_L3D\_CACHE\_HIT, Level 3 data cache hit due to the SME2 unit, event

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

### Related telemetry artifacts

#### Metrics

- [cme\\_l3d\\_cache\\_hit\\_ratio](#) in [CME\\_L3D\\_Cache\\_Effectiveness](#)
- [cme\\_l3d\\_cache\\_hit\\_ratio](#) in [CME\\_Miss\\_Ratio](#)
- [cme\\_system\\_l3d\\_cache\\_hit\\_ratio](#)

#### Metric groups

[CME\\_L3D\\_Cache\\_Effectiveness](#)  
[CME\\_Miss\\_Ratio](#)  
[CME\\_System\\_Memory\\_Effectiveness](#)

#### Functional groups

[CME\\_L3D\\_Cache](#)

### **0x328f CME\_L3D\_CACHE\_MISS, Level 3 data cache demand access miss due to the SME2 unit, event**

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

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_L3D\\_Cache](#)

### **0x3290 CME\_L3D\_CACHE\_REFILL, Level 3 data cache refill due to the SME2 unit, event**

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

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_l3d\\_cache\\_mpki](#) in [CME\\_L3D\\_Cache\\_Effectiveness](#)
- [cme\\_l3d\\_cache\\_mpki](#) in [CME\\_MPKI](#)
- [cme\\_l3d\\_cache\\_miss\\_ratio](#) in [CME\\_L3D\\_Cache\\_Effectiveness](#)
- [cme\\_l3d\\_cache\\_miss\\_ratio](#) in [CME\\_Miss\\_Ratio](#)

##### **Metric groups**

[CME\\_L3D\\_Cache\\_Effectiveness](#)

[CME\\_MPKI](#)

[CME\\_Miss\\_Ratio](#)

##### **Functional groups**

[CME\\_L3D\\_Cache](#)

### **0x3291 CME\_L3D\_CACHE\_REFILL\_PRFM, Level 3 data cache refill due to the SME2 unit, software preload, event**

Counts cacheable reads generated by hardware or software 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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_L3D\\_Cache](#)

### **0x3292 CME\_L3D\_CACHE\_REFILL\_RD, Level 3 data cache refill due to the SME2 unit, read, event**

- This event duplicates L3D\_CACHE\_REFILL.
- If either the core is configured without a per-core L2 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 the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L3D\\_Cache](#)

### 0x32b3 CME\_L3D\_CACHE\_PRFM, Level 3 data cache software preload due to the SME2 unit, event

Counts level 3 cache accesses generated by software prefetches.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L3D\\_Cache](#)

### 0x32b4 CME\_L3D\_CACHE\_HWPRF, SME2 unit Level 3 data cache hardware prefetch, event

Counts level 3 cache accesses generated by hardware prefetches.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_L3D\\_Cache](#)

## 6.4 CME\_LL\_Cache (SME2 LL CACHE) events for C1-SME2

SME2 unit Last Level Cache related events.

Summary of events in CME\_LL\_Cache:

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

**Table 6-4: CME\_LL\_Cache events summary**

Code	Mnemonic	Name	Description
0x3293	<a href="#">CME_LL_CACHE</a>	Last level cache access due to the SME2 unit	Counts transactions that were returned from outside the core cluster. This event counts when the...

Code	Mnemonic	Name	Description
0x3294	<a href="#">CME_LL_CACHE_RD</a>	Last level cache access due to the SME2 unit, read	Counts read transactions that were returned from outside the core cluster. This event counts when...
0x3295	<a href="#">CME_LL_CACHE_MISS_RD</a>	Last level cache miss due to the SME2 unit	Counts read transactions that were returned from outside the core cluster but missed in the...
0x3296	<a href="#">CME_LL_CACHE_HIT</a>	Last level cache hit due to the SME2 unit	Counts transactions that were returned from outside the core cluster, hitting in the last. level...

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### 0x3293 CME\_LL\_CACHE, Last level cache access due to the SME2 unit, event

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

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_LL\\_Cache](#)

### 0x3294 CME\_LL\_CACHE\_RD, Last level cache access due to the SME2 unit, read, event

Counts read transactions that were returned from outside the core cluster. This event counts when the system register CMECFG.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.

This event is a superset of the CME\_LL\_CACHE\_MISS\_RD event.

#### Related telemetry artifacts

##### Metrics

- [cme\\_ll\\_cache\\_read\\_miss\\_ratio](#) in [CME\\_LL\\_Cache\\_Effectiveness](#)
- [cme\\_ll\\_cache\\_read\\_miss\\_ratio](#) in [CME\\_Miss\\_Ratio](#)
- [cme\\_ll\\_cache\\_read\\_hit\\_ratio](#)

##### Metric groups

[CME\\_LL\\_Cache\\_Effectiveness](#)

[CME\\_Miss\\_Ratio](#)

##### Functional groups

[CME\\_LL\\_Cache](#)

### 0x3295 CME\_LL\_CACHE\_MISS\_RD, Last level cache miss due to the SME2 unit, 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 CMECFG.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.

This event is a subset of the CME\_LL\_CACHE\_RD event.

#### Related telemetry artifacts

##### Metrics

- [cme\\_ll\\_cache\\_read\\_mpki](#) in [CME\\_LL\\_Cache\\_Effectiveness](#)
- [cme\\_ll\\_cache\\_read\\_mpki](#) in [CME\\_MPki](#)
- [cme\\_ll\\_cache\\_read\\_miss\\_ratio](#) in [CME\\_LL\\_Cache\\_Effectiveness](#)
- [cme\\_ll\\_cache\\_read\\_miss\\_ratio](#) in [CME\\_Miss\\_Ratio](#)
- [cme\\_ll\\_cache\\_read\\_hit\\_ratio](#)

##### Metric groups

[CME\\_LL\\_Cache\\_Effectiveness](#)  
[CME\\_MPki](#)  
[CME\\_Miss\\_Ratio](#)

##### Functional groups

[CME\\_LL\\_Cache](#)

#### 0x3296 CME\_LL\_CACHE\_HIT, Last level cache hit due to the SME2 unit, event

Counts transactions that were returned from outside the core cluster, hitting in the last. level cache. This event counts when the system register CMECFG.EXTLLC bit is set.

#### Related telemetry artifacts

##### Metrics

- [cme\\_system\\_ll\\_cache\\_hit\\_ratio](#)

##### Metric groups

[CME\\_System\\_Memory\\_Effectiveness](#)

##### Functional groups

[CME\\_LL\\_Cache](#)

## 6.5 CME\_Memory (SME2 MEMORY) events for C1-SME2

SME2 unit Memory system related events.

Summary of events in CME\_Memory:

- Total implemented Common events: 0
- Total Implemented Product ImpDef events: 9
- PMU Only events : 9

- ETE Only events : 0

**Table 6-5: CME\_Memory events summary**

Code	Mnemonic	Name	Description
0x3256	CME_LDST_ALIGN_LAT	SME2 unit access with additional latency from alignment	Counts the number of memory read and write accesses in a cycle that incurred additional latency,...
0x3258	CME_LD_ALIGN_LAT	SME load access with additional latency from alignment	Counts the number of memory read accesses in a cycle that incurred additional latency, due to the...
0x325b	CME_ST_ALIGN_LAT	SME2 unit store access with additional latency from alignment	Counts the number of memory write accesses in a cycle that incurred additional latency, due to...
0x3298	CME_MEM_ACCESS	SME2 unit Data memory access	Counts memory accesses issued by the SME2 unit, where those accesses are issued due to load or...
0x3299	CME_MEM_ACCESS_RD	SME2 unit Data memory access, read	Counts memory accesses issued by the CPU due to load operations. The event counts any memory load...
0x329a	CME_MEM_ACCESS_RD_PERCYC	Number of outstanding memory reads per cycle in the SME2 unit	Counts the number of outstanding loads or memory read accesses per cycle.
0x329b	CME_MEM_ACCESS_WR	SME2 unit Data memory access, write	Counts memory accesses issued by the SME2 unit due to store operations. The event counts any...
0x329c	CME_REMOTE_ACCESS	Access to another socket in a multi-socket system due to the SME2 unit	Counts accesses to another chip, which is implemented as a different CMN mesh in the system. If...
0x329d	CME_REMOTE_ACCESS_RD	Access to another socket in a multi-socket system due to the SME2 unit, read	Counts read accesses to another chip, which is implemented as a different CMN mesh in the system....

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

#### **0x3256 CME\_LDST\_ALIGN\_LAT, SME2 unit access with additional latency from alignment, event**

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

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Memory](#)

#### **0x3258 CME\_LD\_ALIGN\_LAT, SME load access with additional latency from alignment, event**

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

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Memory](#)

### **0x325b CME\_ST\_ALIGN\_LAT, SME2 unit store access with additional latency from alignment, event**

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

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Memory](#)

### **0x3298 CME\_MEM\_ACCESS, SME2 unit Data memory access, event**

Counts memory accesses issued by the SME2 unit, where those accesses are issued due to load or store operations. This event counts memory accesses no matter whether the data is received from any level of cache hierarchy or external memory. If memory accesses are broken up into smaller transactions than what were specified in the load or store instructions, then the event counts those smaller memory transactions.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Memory](#)

### **0x3299 CME\_MEM\_ACCESS\_RD, SME2 unit Data memory access, read, event**

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

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Memory](#)

### **0x329a CME\_MEM\_ACCESS\_RD\_PERCYC, Number of outstanding memory reads per cycle in the SME2 unit, event**

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

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Memory](#)

### **0x329b CME\_MEM\_ACCESS\_WR, SME2 unit Data memory access, write, event**

Counts memory accesses issued by the SME2 unit 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. If memory accesses are broken up by the load/store unit into smaller transactions that are issued by the bus interface, then the event counts those smaller transactions.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Memory](#)

### **0x329c CME\_REMOTE\_ACCESS, Access to another socket in a multi-socket system due to the SME2 unit, event**

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

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Memory](#)

### **0x329d CME\_REMOTE\_ACCESS\_RD, Access to another socket in a multi-socket system due to the SME2 unit, 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 the C1-SME2 Telemetry Specification.

## Functional groups

[CME\\_Memory](#)

## 6.6 CME\_Retired (SME2 RETIRED) events for C1-SME2

SME2 unit Retired instruction and operation events.

Summary of events in CME\_Retired:

- Total implemented Common events: 2
- Total Implemented Product ImpDef events: 2
- PMU Only events : 4
- ETE Only events : 0

**Table 6-6: CME\_Retired events summary**

Code	Mnemonic	Name	Description
0x3247	<a href="#">CME_INST_RETIRED</a>	SME Instruction architecturally executed	Counts instructions that have been architecturally executed inside the SME2 unit.
0x3248	<a href="#">CME_OP_RETIRED</a>	SME Micro-operation architecturally executed	Counts micro-operations that are architecturally executed. This is a count of number of...
0x8	<a href="#">INST_RETIRED</a>	Operation retired	Counts instructions that have been architecturally executed.
0x8000	<a href="#">SIMD_INST_RETIRED</a>	Operation retired, SIMD, including load and store	The counter counts each retired SIMD instruction counted by INST_RETIRED. This counter does not...

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### 0x3247 CME\_INST\_RETIRED, SME Instruction architecturally executed, event

Counts instructions that have been architecturally executed inside the SME2 unit.

#### Related telemetry artifacts

##### Metrics

- [cme\\_ipc](#)
- [cme\\_l1d\\_cache\\_mpki](#) in [CME\\_L1D\\_Cache\\_Effectiveness](#)
- [cme\\_l1d\\_cache\\_mpki](#) in [CME\\_MPKI](#)
- [cme\\_l3d\\_cache\\_mpki](#) in [CME\\_L3D\\_Cache\\_Effectiveness](#)
- [cme\\_l3d\\_cache\\_mpki](#) in [CME\\_MPKI](#)
- [cme\\_ll\\_cache\\_read\\_mpki](#) in [CME\\_LL\\_Cache\\_Effectiveness](#)
- [cme\\_ll\\_cache\\_read\\_mpki](#) in [CME\\_MPKI](#)

##### Metric groups

[CME\\_General](#)

CME\_L1D\_Cache\_Effectiveness  
CME\_L3D\_Cache\_Effectiveness  
CME\_LL\_Cache\_Effectiveness  
CME\_MPKI

#### Functional groups

CME\_Retired

### 0x3248 CME\_OP\_RETIRE, SME 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

- cme\_load\_percentage
- cme\_store\_percentage
- streaming\_ld\_op\_percentage
- streaming\_st\_op\_percentage
- streaming\_ls\_op\_percentage

##### Metric groups

CME\_Operation\_Mix

#### Functional groups

CME\_Retired

### 0x8 INST\_RETIRE, Operation retired, event

Counts instructions that have been architecturally executed.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

CME\_Retired

### 0x8000 SIMD\_INST\_RETIRE, Operation retired, SIMD, including load and store, event

The counter counts each retired SIMD instruction counted by INST\_RETIRE.

This counter does not count scalar instructions counted by ASE\_INST\_RETIRE.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

CME\_Retired

## 6.7 CME\_Spec\_Operation (SME2 SPEC OPERATION) events for C1-SME2

SME2 unit Speculatively executed operations related events.

Summary of events in CME\_Spec\_Operation:

- Total implemented Common events: 38
- Total Implemented Product ImpDef events: 29
- PMU Only events : 67
- ETE Only events : 0

**Table 6-7: CME\_Spec\_Operation events summary**

Code	Mnemonic	Name	Description
0x1b	INST_SPEC	Operation speculatively executed	Counts instructions that have been speculatively executed.
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.
0x321b	SSVE_LDST_SPEC	Operation speculatively executed, Streaming SVE load, store, or prefetch	The counter counts each Load or Store operation counted by SVE_LDST_SPEC when CPU executes in...
0x321c	SSVE_LD_SPEC	Operation speculatively executed, Streaming SVE load	The counter counts each Load operation counted by SVE_LD_SPEC when CPU executes in Streaming mode
0x321d	SSVE_ST_SPEC	Operation speculatively executed, Streaming SVE store	The counter counts each Store operation counted by SVE_ST_SPEC when CPU executes in Streaming mode
0x321f	SSVE_INT_SPEC	Operation speculatively executed, Streaming SVE integer	The counter counts each Speculatively executed integer arithmetic operation due to an...
0x3221	SSVE_INT8_SPEC	Operation speculatively executed, Streaming SVE 8-bit integer	The counter counts each operation counted by SSVE_INT_SPEC where the largest type is 8-bit integer.
0x3222	SSVE_INT16_SPEC	Operation speculatively executed, Streaming SVE 16-bit integer	The counter counts each operation counted by SSVE_INT_SPEC where the largest type is 16-bit integer.
0x3223	SSVE_INT32_SPEC	Operation speculatively executed, Streaming SVE 32-bit integer	The counter counts each operation counted by SSVE_INT_SPEC where the largest type is 32-bit integer.
0x3224	SSVE_INT64_SPEC	Operation speculatively executed, Streaming SVE 64-bit integer	The counter counts each operation counted by SSVE_INT_SPEC where the largest type is 64-bit integer.

Code	Mnemonic	Name	Description
0x3229	SSVE_INT_MUL_SPEC	Operation speculatively executed, Streaming SVE integer multiply or multiply-accumulate	The counter counts each Speculatively executed integer multiply or multiply-accumulate...
0x322a	SSVE_INT_DOT_SPEC	Operation speculatively executed, Streaming SVE integer dot product	The counter counts each integer dot product operation counted by ASE_SVE_INT_DOT_SPEC due to any...
0x322b	SSVE_FP_ADDSUB_SPEC	Operation speculatively executed, Streaming SVE floating-point add or subtract	The counter counts each Speculatively executed floating-point add or subtract operation counted...
0x322c	SSVE_FP_MUL_SPEC	Operation speculatively executed, Streaming SVE floating-point multiply	The counter counts each Speculatively executed floating-point multiply operation counted...
0x322d	SSVE_FP_FMA_SPEC	Operation speculatively executed, Streaming SVE floating-point multiply-add or multiply-subtract	The counter counts each Speculatively executed floating-point fused multiply-add...
0x322e	SSVE_FP_DOT_SPEC	Operation speculatively executed, Streaming SVE floating-point dot product	The counter counts each dot-product operation counted by SSVE_FP_SPEC due to any of the following...
0x322f	SSVE_FP_SQRT_SPEC	Floating-point operation speculatively executed, Streaming SVE square root	The counter counts each Speculatively executed floating-point square-root operation counted...
0x3230	SSVE_FP_DIV_SPEC	Floating-point operation speculatively executed, Streaming SVE divide	The counter counts each Speculatively executed floating-point divide operation counted...
0x3231	SSVE_FP_RECPE_SPEC	Floating-point operation speculatively executed, Streaming SVE reciprocal estimate	The counter counts each Speculatively executed floating-point reciprocal estimate...
0x3232	SSVE_FP_CVT_SPEC	Floating-point operation speculatively executed, Streaming SVE convert	The counter counts each Speculatively executed floating-point convert operation counted...
0x3233	SSVE_FP_VREDUCE_SPEC	Floating-point operation speculatively executed, Streaming SVE vector reduction	The counter counts each Speculatively executed floating-point tree-wise reduction operation...
0x3253	CME_LDST_SPEC	SME Operation speculatively executed, load or store	Counts load and store operations that have been speculatively executed.
0x3254	CME_LD_SPEC	SME Operation speculatively executed, load	Counts speculatively executed SME load operations including Single Instruction Multiple Data...
0x3255	CME_UNALIGNED_LDST_SPEC	SME Operation speculatively executed, unaligned load or store	Counts unaligned memory operations issued by the CPU. This event counts unaligned accesses (as...
0x3257	CME_UNALIGNED_LD_SPEC	SME Operation speculatively executed, unaligned load	Counts unaligned memory read operations issued by the CPU. This event counts unaligned accesses...
0x3259	CME_ST_SPEC	SME Operation speculatively executed, store	Counts speculatively executed SME store operations including Single Instruction Multiple Data...
0x325a	CME_UNALIGNED_ST_SPEC	SME Operation speculatively executed, unaligned store	Counts unaligned memory write operations issued by the CPU. This event counts unaligned accesses...

Code	Mnemonic	Name	Description
0x325c	CME_PRF_SPEC	SME operation speculatively executed, Prefetch	Counts speculatively executed operations in Streaming SVE mode that prefetch memory, executed by...
0x32af	SME_INT_OTHER_SPEC	Operation speculatively executed, other SME integer	The counter counts each speculatively executed operation counted by SME_INT_SPEC due to an...
0x74	ASE_SPEC	Operation speculatively executed, Advanced SIMD	The counter counts each operation counted by INST_SPEC that is an Advanced SIMD...
0x8004	SIMD_INST_SPEC	Operation speculatively executed, SIMD, including load and store	The counter counts each speculatively executed SIMD operation counted by INST_SPEC. This counter...
0x8040	INT_SPEC	Integer Operation speculatively executed	Counts integer operations that have been speculatively executed.
0x8043	ASE_SVE_INT_SPEC	Operation speculatively executed, Advanced SIMD or SVE integer	The counter counts each Advanced SIMD or SVE operation counted by INT_SPEC where the type...
0x8056	SVE_SPEC	Operation speculatively executed, SVE	The counter counts each operation counted by ASE_SVE_SPEC that is a scalable vector data...
0x8057	ASE_SVE_SPEC	Operation speculatively executed, Advanced SIMD or SVE	The counter counts each operation counted by SE_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 or store operation counted by any of...
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...
0x80e3	ASE_SVE_INT8_SPEC	Operation speculatively executed, Advanced SIMD or SVE 8-bit integer	The counter counts each operation counted by ASE_SVE_INT_SPEC where the largest type is...
0x80e7	ASE_SVE_INT16_SPEC	Operation speculatively executed, Advanced SIMD or SVE 16-bit integer	The counter counts each operation counted by ASE_SVE_INT_SPEC where the largest type is...
0x80eb	ASE_SVE_INT32_SPEC	Operation speculatively executed, Advanced SIMD or SVE 32-bit integer	The counter counts each operation counted by ASE_SVE_INT_SPEC where the largest type is...
0x80ef	ASE_SVE_INT64_SPEC	Operation speculatively executed, Advanced SIMD or SVE 64-bit integer	The counter counts each operation counted by ASE_SVE_INT_SPEC where the largest type is...
0x8352	SME_FP_SPEC	Operation speculatively executed, SME floating-point	The counter counts each speculatively executed floating-point operation counted by SE_SPEC due to...

Code	Mnemonic	Name	Description
0x835c	SME_SPEC	Operation speculatively executed, SME data processing	The counter counts each operation counted by SE_SPEC that is an SME data-processing...
0x835d	SE_SPEC	Operation speculatively executed, Advanced SIMD, scalable vector extension, or scalable matrix extension 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...
0x8360	SME_INT8_SPEC	Operation speculatively executed, SME 8-bit integer	The counter counts each speculatively executed 8-bit integer operation counted by SME_INT_SPEC...
0x8364	SME_INT16_SPEC	Operation speculatively executed, SME 16-bit integer	The counter counts each speculatively executed 16-bit integer operation counted by SME_INT_SPEC...
0x8366	SME_FP_HP_SPEC	Operation speculatively executed, SME half-precision floating-point	The counter counts each speculatively executed half-precision floating-point operation counted...
0x8378	SME_INT_SPEC	Operation speculatively executed, SME integer	The counter counts each speculatively executed integer operation counted by SE_INT_SPEC due to...
0x837a	SME_INT_MUL_SPEC	Operation speculatively executed, SME integer multiply or multiply-accumulate	The counter counts each speculatively executed integer multiply, multiply-add, or...
0x837c	SME_INT_DOT_SPEC	Operation speculatively executed, SME integer dot product	The counter counts each speculatively executed integer dot product operation counted...
0x837e	SME_INT_MOPA_SPEC	Operation speculatively executed, SME integer outer product and accumulate, or outer product and subtract	The counter counts each speculatively executed integer outer product and accumulate, or outer...
0x8388	SME_LDST_ZAREG_SPEC	SME ZA unpredicated load/store	The counter counts each speculatively executed operation that reads from or writes to memory...
0x8389	SME_LD_ZAREG_SPEC	SME ZA unpredicated load.	The counter counts each speculatively executed operation that reads from memory counted...
0x838a	SME_ST_ZAREG_SPEC	SME ZA unpredicated store	The counter counts each speculatively executed operation that writes to memory counted...
0x838c	SME_LDST_ZTREG_SPEC	SME ZT unpredicated load/store	The counter counts each speculatively executed operation that reads from or writes to memory...
0x838d	SME_LD_ZTREG_SPEC	SME ZT unpredicated load	The counter counts each speculatively executed operation that reads from memory counted...
0x838e	SME_ST_ZTREG_SPEC	SME ZT unpredicated store	The counter counts each speculatively executed operation that writes to memory counted...
0x8390	SME_LDST_REG_SPEC	SME unpredicated load/store	The counter counts each speculatively executed operation that reads from or writes to memory...

Code	Mnemonic	Name	Description
0x8391	<a href="#">SME_LD_REG_SPEC</a>	SME unpredicated load	The counter counts each speculatively executed operation that reads from memory counted...
0x8392	<a href="#">SME_ST_REG_SPEC</a>	SME unpredicated store	The counter counts each speculatively executed operation that writes to memory counted...
0x8394	<a href="#">SME_LDST_TILE_SPEC</a>	SME predicated tile load/store	The counter counts each speculatively executed operation that reads from or writes to memory...
0x8395	<a href="#">SME_LD_TILE_SPEC</a>	SME predicated tile load	The counter counts each speculatively executed operation that reads from memory counted...
0x8396	<a href="#">SME_ST_TILE_SPEC</a>	SME predicated tile store	The counter counts each speculatively executed operation that writes to memory counted...
0x839a	<a href="#">SME_LUT_SPEC</a>	Lookup table operation speculatively executed, SME	The counter counts each speculatively executed LUT operation counted by SME_SPEC that returns...

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### 0x1b INST\_SPEC, Operation speculatively executed, event

Counts instructions that have been speculatively executed.

#### Related telemetry artifacts

##### Metrics

- [streaming\\_op\\_percentage](#)
- [za\\_op\\_percentage](#)

##### Metric groups

[CME\\_Operation\\_Mix](#)

##### Functional groups

[CME\\_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

##### Metrics

- [streaming\\_op\\_percentage](#)
- [streaming\\_int\\_op\\_percentage](#)
- [streaming\\_fp\\_op\\_percentage](#)
- [streaming\\_slow\\_inst\\_percentage](#)

### Metric groups

[CME\\_Operation\\_Mix](#)

### Functional groups

[CME\\_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 the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Spec\\_Operation](#)

## **0x321b SSVE\_LDST\_SPEC, Operation speculatively executed, Streaming SVE load, store, or prefetch, event**

The counter counts each Load or Store operation counted by `SVE_LDST_SPEC` when CPU executes in Streaming mode

### Related telemetry artifacts

#### Metrics

- [streaming\\_ls\\_op\\_percentage](#)

### Metric groups

[CME\\_Operation\\_Mix](#)

### Functional groups

[CME\\_Spec\\_Operation](#)

## **0x321c SSVE\_LD\_SPEC, Operation speculatively executed, Streaming SVE load, event**

The counter counts each Load operation counted by `SVE_LD_SPEC` when CPU executes in Streaming mode

### Related telemetry artifacts

#### Metrics

- [streaming\\_ld\\_op\\_percentage](#)

### Metric groups

[CME\\_Operation\\_Mix](#)

### Functional groups

[CME\\_Spec\\_Operation](#)

## **0x321d SSVE\_ST\_SPEC, Operation speculatively executed, Streaming SVE store, event**

The counter counts each Store operation counted by `SVE_ST_SPEC` when CPU executes in Streaming mode

## Related telemetry artifacts

### Metrics

- [streaming\\_st\\_op\\_percentage](#)

### Metric groups

[CME\\_Operation\\_Mix](#)

### Functional groups

[CME\\_Spec\\_Operation](#)

## **0x321f SSVE\_INT\_SPEC, Operation speculatively executed, Streaming SVE integer, event**

The counter counts each Speculatively executed integer arithmetic operation due to an SVE data-processing instruction listed in SVE integer instructions.

This counter only counts operations executed in Streaming mode.

## Related telemetry artifacts

### Metrics

- [streaming\\_int\\_op\\_percentage](#)

### Metric groups

[CME\\_Operation\\_Mix](#)

### Functional groups

[CME\\_Spec\\_Operation](#)

## **0x3221 SSVE\_INT8\_SPEC, Operation speculatively executed, Streaming SVE 8-bit integer, event**

The counter counts each operation counted by SSVE\_INT\_SPEC where the largest type is 8-bit integer.

## Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Spec\\_Operation](#)

## **0x3222 SSVE\_INT16\_SPEC, Operation speculatively executed, Streaming SVE 16-bit integer, event**

The counter counts each operation counted by SSVE\_INT\_SPEC where the largest type is 16-bit integer.

## Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x3223 SSVE\_INT32\_SPEC, Operation speculatively executed, Streaming SVE 32-bit integer, event**

The counter counts each operation counted by SSVE\_INT\_SPEC where the largest type is 32-bit integer.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x3224 SSVE\_INT64\_SPEC, Operation speculatively executed, Streaming SVE 64-bit integer, event**

The counter counts each operation counted by SSVE\_INT\_SPEC where the largest type is 64-bit integer.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x3229 SSVE\_INT\_MUL\_SPEC, Operation speculatively executed, Streaming SVE integer multiply or multiply-accumulate, event**

The counter counts each Speculatively executed integer multiply or multiply-accumulate operation counted by SSVE\_INT\_SPEC, specifically in Streaming mode, due to any of the following instructions:

- SVE: MAD, MLA, MLS, MSB, MUL, SMULH, or UMULH.
- SVE2: CMLA, MLA, MLS, MUL, PMUL, SMLALB, SMLALT, SMLSLB, SMLSLT, SMULH, SMULLB, SMULLT, SQDMLALB, SQDMLALBT, SQDMLALT, SQDMLSLB, SQDMLSLBT, SQDMLSLT, SQDMULH, SQDMULLB, SQDMULLT, SQRDCMLAH, SQRDMLAH, SQRDMLSH, SQRDMULH, UMLALB, UMLALT, UMLSLB, UMLSLT, UMULH, UMULLB, or UMULLT.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x322a SSVE\_INT\_DOT\_SPEC, Operation speculatively executed, Streaming SVE integer dot product, event**

The counter counts each integer dot product operation counted by ASE\_SVE\_INT\_DOT\_SPEC due to any of the following instructions:

- SVE: SDOT, SUDOT, UDOT, or USDOT.

- SVE2: CDOT, SUDOT, or USDOT.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

#### **0x322b SSVE\_FP\_ADDSUB\_SPEC, Operation speculatively executed, Streaming SVE floating-point add or subtract, event**

The counter counts each Speculatively executed floating-point add or subtract operation counted by SSVE\_FP\_SPEC due to any of the following instructions:

- SVE: FABD, FADD, FSUB, or FSUBR.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

#### **0x322c SSVE\_FP\_MUL\_SPEC, Operation speculatively executed, Streaming SVE floating-point multiply, event**

The counter counts each Speculatively executed floating-point multiply operation counted by SSVE\_FP\_SPEC due to any of the following instructions:

- SVE: FMUL, FMULX, or FTSMUL.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

#### **0x322d SSVE\_FP\_FMA\_SPEC, Operation speculatively executed, Streaming SVE floating-point multiply-add or multiply-subtract, event**

The counter counts each Speculatively executed floating-point fused multiply-add or multiply-subtract operation counted by FP\_FMA\_SPEC, specifically in Streaming mode, due to any of the following instructions:

- SVE: BFMLALB (vectors), BFMLALT (vectors), FCMLA (vectors), FMAD, FMLA (vectors), FMLS (vectors), FMSB, FNMAD, FNMLA, FNMLS, FNMSB, or FTMAD.
- SVE2: BFMLALB (vectors), BFMLALT (vectors), FMLALB (vectors), FMLALT (vectors), FMLSBLB (vectors), or FMLSLLT (vectors).

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x322e SSVE\_FP\_DOT\_SPEC, Operation speculatively executed, Streaming SVE floating-point dot product, event**

The counter counts each dot-product operation counted by SSVE\_FP\_SPEC due to any of the following instructions:

- SVE: BFDOT.
- SVE2: BFDOT.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x322f SSVE\_FP\_SQRT\_SPEC, Floating-point operation speculatively executed, Streaming SVE square root, event**

The counter counts each Speculatively executed floating-point square-root operation counted by SSVE\_FP\_SPEC, specifically in Streaming mode, due to any of the following instructions:

- SVE: FSQRT.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x3230 SSVE\_FP\_DIV\_SPEC, Floating-point operation speculatively executed, Streaming SVE divide, event**

The counter counts each Speculatively executed floating-point divide operation counted by FP\_DIV\_SPEC, specifically in Streaming mode, due to any of the following instructions:

- SVE: FDIV or FDIVR.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x3231 SSVE\_FP\_RECPE\_SPEC, Floating-point operation speculatively executed, Streaming SVE reciprocal estimate, event**

The counter counts each Speculatively executed floating-point reciprocal estimate operation counted by SSVE\_FP\_SPEC, specifically in Streaming mode, due to any of the following instructions:

- SVE: FRECPE or FRSQRTE.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x3232 SSVE\_FP\_CVT\_SPEC, Floating-point operation speculatively executed, Streaming SVE convert, event**

The counter counts each Speculatively executed floating-point convert operation counted by SSVE\_FP\_SPEC, specifically in Streaming mode, due to an SVE instruction.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x3233 SSVE\_FP\_VREDUCE\_SPEC, Floating-point operation speculatively executed, Streaming SVE vector reduction, event**

The counter counts each Speculatively executed floating-point treewise reduction operation counted by SSVE\_FP\_SPEC, specifically in Streaming mode, due to any of the following A64 instructions:

- SVE: FADDV, FMAXNMV, FMAXV, FMINNVMV, or FMINV.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x3253 CME\_LDST\_SPEC, SME Operation speculatively executed, load or store, event**

Counts load and store operations that have been speculatively executed.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

## Functional groups

[CME\\_Spec\\_Operation](#)

### 0x3254 CME\_LD\_SPEC, SME Operation speculatively executed, load, event

Counts speculatively executed SME load operations including Single Instruction Multiple Data (SIMD) load operations executed in streaming SVE mode.

#### Related telemetry artifacts

##### Metrics

- [cme\\_load\\_percentage](#)

##### Metric groups

[CME\\_Operation\\_Mix](#)

##### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x3255 CME\_UNALIGNED\_LDST\_SPEC, SME Operation speculatively executed, unaligned load or store, event

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

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

##### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x3257 CME\_UNALIGNED\_LD\_SPEC, SME Operation speculatively executed, unaligned load, event

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

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

##### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x3259 CME\_ST\_SPEC, SME Operation speculatively executed, store, event

Counts speculatively executed SME store operations including Single Instruction Multiple Data (SIMD) load operations executed in streaming SVE mode.

#### Related telemetry artifacts

##### Metrics

- [cme\\_store\\_percentage](#)

### Metric groups

[CME\\_Operation\\_Mix](#)

### Functional groups

[CME\\_Spec\\_Operation](#)

## **0x325a CME\_UNALIGNED\_ST\_SPEC, SME Operation speculatively executed, unaligned store, event**

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

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Spec\\_Operation](#)

## **0x325c CME\_PRF\_SPEC, SME operation speculatively executed, Prefetch, event**

Counts speculatively executed operations in Streaming SVE mode that prefetch memory, executed by the SME2 unit.

Only the SME RPRFM instructions are counted by this event.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Spec\\_Operation](#)

## **0x32af SME\_INT\_OTHER\_SPEC, Operation speculatively executed, other SME integer, event**

The counter counts each speculatively executed operation counted by SME\_INT\_SPEC due to an instruction which reads from or writes to any part of the ZA array, which is not counted by SME\_INT\_DOT\_SPEC, SME\_INT\_MOPA\_SPEC or SME\_INT\_MUL\_SPEC

### Related telemetry artifacts

#### Metrics

- [za\\_int\\_other\\_percentage](#)

### Metric groups

[CME\\_Operation\\_Mix](#)

### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x74 ASE\_SPEC, Operation speculatively executed, Advanced SIMD, 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**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8004 SIMD\_INST\_SPEC, Operation speculatively executed, SIMD, including load and store, event**

The counter counts each speculatively executed SIMD operation counted by INST\_SPEC.

This counter does not count scalar operations counted by ASE\_INST\_SPEC.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8040 INT\_SPEC, Integer Operation speculatively executed, event**

Counts integer operations that have been speculatively executed.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8043 ASE\_SVE\_INT\_SPEC, Operation speculatively executed, Advanced SIMD or SVE integer, event**

The counter counts each Advanced SIMD or SVE operation counted by INT\_SPEC where the type is integer.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8056 SVE\_SPEC, Operation speculatively executed, SVE, event**

The counter counts each operation counted by ASE\_SVE\_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

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x8057 ASE\_SVE\_SPEC, Operation speculatively executed, Advanced SIMD or SVE, event**

The counter counts each operation counted by SE\_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 the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x8080 SVE\_LDST\_SPEC, Operation speculatively executed, SVE load, store, or prefetch, event**

The counter counts each Speculatively executed load or store operation counted by any of SVE\_LD\_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 the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x80e3 ASE\_SVE\_INT8\_SPEC, Operation speculatively executed, Advanced SIMD or SVE 8-bit integer, event**

The counter counts each operation counted by ASE\_SVE\_INT\_SPEC where the largest type is 8-bit integer.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x80e7 ASE\_SVE\_INT16\_SPEC, Operation speculatively executed, Advanced SIMD or SVE 16-bit integer, event**

The counter counts each operation counted by ASE\_SVE\_INT\_SPEC where the largest type is 16-bit integer.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x80eb ASE\_SVE\_INT32\_SPEC, Operation speculatively executed, Advanced SIMD or SVE 32-bit integer, event**

The counter counts each operation counted by ASE\_SVE\_INT\_SPEC where the largest type is 32-bit integer.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x80ef ASE\_SVE\_INT64\_SPEC, Operation speculatively executed, Advanced SIMD or SVE 64-bit integer, event**

The counter counts each operation counted by ASE\_SVE\_INT\_SPEC where the largest type is 64-bit integer.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x8352 SME\_FP\_SPEC, Operation speculatively executed, SME floating-point, event**

The counter counts each speculatively executed floating-point operation counted by SE\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

### Related telemetry artifacts

#### Metrics

- [za\\_fp\\_op\\_percentage](#)
- [za\\_fp\\_mopa\\_percentage](#)
- [za\\_fp\\_fma\\_percentage](#)
- [za\\_fp\\_dot\\_percentage](#)
- [za\\_fp\\_addsub\\_percentage](#)
- [za\\_fp\\_other\\_percentage](#)

#### Metric groups

[CME\\_Operation\\_Mix](#)

#### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x835c SME\_SPEC, Operation speculatively executed, SME data processing, event**

The counter counts each operation counted by SE\_SPEC that is an SME data-processing operation. Operations due to the following instructions are counted as SME data-processing operations:

- Data-processing 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

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x835d SE\_SPEC, Operation speculatively executed, Advanced SIMD, scalable vector extension, or scalable matrix extension 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 the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_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

- [za\\_op\\_percentage](#)
- [za\\_int\\_op\\_percentage](#)
- [za\\_fp\\_op\\_percentage](#)

#### Metric groups

[CME\\_Operation\\_Mix](#)

#### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x8360 SME\_INT8\_SPEC, Operation speculatively executed, SME 8-bit integer, event**

The counter counts each speculatively executed 8-bit integer operation counted by SME\_INT\_SPEC due to an instruction which reads from or writes to any part of the ZA or ZT array.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x8364 SME\_INT16\_SPEC, Operation speculatively executed, SME 16-bit integer, event

The counter counts each speculatively executed 16-bit integer operation counted by SME\_INT\_SPEC due to an instruction which reads from or writes to any part of the ZA or ZT array.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x8366 SME\_FP\_HP\_SPEC, Operation speculatively executed, SME half-precision floating-point, event

The counter counts each speculatively executed half-precision floating-point operation counted by SME\_FP\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x8378 SME\_INT\_SPEC, Operation speculatively executed, SME integer, event

The counter counts each speculatively executed integer operation counted by SE\_INT\_SPEC due to an instruction which reads from or writes to any part of the ZA or ZT array.

### Related telemetry artifacts

#### Metrics

- [za\\_int\\_op\\_percentage](#)
- [za\\_int\\_mopa\\_percentage](#)
- [za\\_int\\_dot\\_percentage](#)
- [za\\_int\\_other\\_percentage](#)

#### Metric groups

[CME\\_Operation\\_Mix](#)

#### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x837a SME\_INT\_MUL\_SPEC, Operation speculatively executed, SME integer multiply or multiply-accumulate, event**

The counter counts each speculatively executed integer multiply, multiply-add, or multiply-subtract operation counted by SE\_INT\_MUL\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x837c SME\_INT\_DOT\_SPEC, Operation speculatively executed, SME integer dot product, event**

The counter counts each speculatively executed integer dot product operation counted by SE\_INT\_DOT\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

#### **Related telemetry artifacts**

##### **Metrics**

- [za\\_int\\_dot\\_percentage](#)

##### **Metric groups**

[CME\\_Operation\\_Mix](#)

##### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x837e SME\_INT\_MOPA\_SPEC, Operation speculatively executed, SME integer outer product and accumulate, or outer product and subtract, event**

The counter counts each speculatively executed integer outer product and accumulate, or outer product and subtract operation counted by SME\_INT\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

#### **Related telemetry artifacts**

##### **Metrics**

- [za\\_int\\_mopa\\_percentage](#)

##### **Metric groups**

[CME\\_Operation\\_Mix](#)

##### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8388 SME\_LDST\_ZAREG\_SPEC, SME ZA unpredicated load/store, event**

The counter counts each speculatively executed operation that reads from or writes to memory counted by SME\_LDST\_REG\_SPEC that was due to an unpredicated instruction targeting the ZA array

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x8389 SME\_LD\_ZAREG\_SPEC, SME ZA unpredicated load, event

The counter counts each speculatively executed operation that reads from memory counted by SME\_LDST\_ZAREG\_SPEC that was due to an unpredicated instruction targeting the ZA array

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x838a SME\_ST\_ZAREG\_SPEC, SME ZA unpredicated store, event

The counter counts each speculatively executed operation that writes to memory counted by SME\_LDST\_ZAREG\_SPEC that was due to an unpredicated instruction targeting the ZA array.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x838c SME\_LDST\_ZTREG\_SPEC, SME ZT unpredicated load/store, event

The counter counts each speculatively executed operation that reads from or writes to memory counted by SME\_LDST\_REG\_SPEC that was due to an unpredicated instruction targeting the ZT register

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### 0x838d SME\_LD\_ZTREG\_SPEC, SME ZT unpredicated load, event

The counter counts each speculatively executed operation that reads from memory counted by SME\_LDST\_ZTREG\_SPEC that was due to an unpredicated instruction targeting the ZT register

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Spec\\_Operation](#)

### **0x838e SME\_ST\_ZTREG\_SPEC, SME ZT unpredicated store, event**

The counter counts each speculatively executed operation that writes to memory counted by SME\_LDST\_ZTREG\_SPEC that was due to an unpredicated instruction targeting the ZT register.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8390 SME\_LDST\_REG\_SPEC, SME unpredicated load/store, event**

The counter counts each speculatively executed operation that reads from or writes to memory counted by SME\_SPEC that was due to an unpredicated instruction targeting the ZA array or ZT register.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8391 SME\_LD\_REG\_SPEC, SME unpredicated load, event**

The counter counts each speculatively executed operation that reads from memory counted by SME\_LDST\_REG\_SPEC that was due to an unpredicated instruction targeting the ZA array or ZT register.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8392 SME\_ST\_REG\_SPEC, SME unpredicated store, event**

The counter counts each speculatively executed operation that writes to memory counted by SME\_LDST\_REG\_SPEC that was due to an unpredicated instruction targeting the ZA array or ZT register.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8394 SME\_LDST\_TILE\_SPEC, SME predicated tile load/store, event**

The counter counts each speculatively executed operation that reads from or writes to memory counted by SME\_LDST\_REG\_SPEC that was due to an instruction with at least one governing predicate which is targeting the ZA array.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8395 SME\_LD\_TILE\_SPEC, SME predicated tile load, event**

The counter counts each speculatively executed operation that reads from memory counted by SME\_LDST\_TILE\_SPEC that was due to an instruction with at least one governing predicate which is targeting the ZA array.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x8396 SME\_ST\_TILE\_SPEC, SME predicated tile store, event**

The counter counts each speculatively executed operation that writes to memory counted by SME\_LDST\_TILE\_SPEC that was due to an instruction with at least one governing predicate which is targeting the ZA array.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

### **0x839a SME\_LUT\_SPEC, Lookup table operation speculatively executed, SME, event**

The counter counts each speculatively executed LUT operation counted by SME\_SPEC that returns a value from a lookup table made up of a ZT register which is indexed by values from a Z register

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Spec\\_Operation](#)

## 6.8 CME\_Stall (SME2 STALL) events for C1-SME2

SME2 unit Stall related events.

Summary of events in CME\_Stall:

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

**Table 6-8: CME\_Stall events summary**

Code	Mnemonic	Name	Description
0x3249	CME_STALL	No operation sent for execution inside SME2 unit	Counts cycles when no operations are sent in the SME2 unit to the rename from the frontend or...
0x324a	CME_STALL_FRONTEND	No operation sent for execution due to the SME2 unit frontend	No operation has been issued, because of the frontend. The counter counts on any cycle when no...
0x324b	CME_STALL_FRONTEND_CPU	SME2 unit front-end stall due to arbitrated core not sending instructions	The counter counts cycles when CME_STALL_FRONTEND increments, due to not enough instructions...
0x324c	CME_STALL_FRONTEND_OTHER_CPU	SME2 unit front-end stall due to non-arbitrated cores	The counter counts cycles when CME_STALL_FRONTEND increments, due to instructions received...
0x324d	CME_STALL_BACKEND	No operation sent for execution in the SME2 unit due to the backend	Counts cycles whenever the rename unit in the SME2 unit is unable to send any micro-operations to...
0x324e	CME_STALL_BACKEND_CORE	SME2 unit backend stall due to the execution units	Counts cycles when CME_STALL_BACKEND is set, due to DP issue queues not accepting instructions.
0x324f	CME_STALL_BACKEND_MEM	SME2 unit backend stall due to the memory system	Counts cycles when CME_STALL_BACKEND is set, due to Load-Store issue queues not accepting...
0x3250	CME_STALL_BACKEND_PF	SME2 unit backend stall due to the prefetcher	Counts cycles when CME_STALL_BACKEND is set, due to prefetcher issue queues not accepting...
0x3251	CME_STALL_BACKEND_MEM_CACHE	SME2 unit memory system stall due to the cache pipelines	Counts cycles when CME_STALL_BACKEND_MEM is set, with the oldest instruction in at least one of...
0x3252	CME_STALL_BACKEND_MEM_STORE	SME2 unit memory system stall due to store backpressure	Counts cycles when CME_STALL_BACKEND_MEM is set, with the oldest instruction in at least one of...
0x32a8	SSVE_CONTEXT_SWITCH	Context switch stall cycles	Cycles inside the SME2 unit lost for context switch (new instructions are stalled in decode...
0x32a9	SSVE_FAST_CONTEXT_SWITCH	Stall cycles due to another Core accessing its context	Cycles inside the SME2 unit lost because a core loads or stores its context (without causing a...

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### **0x3249 CME\_STALL, No operation sent for execution inside SME2 unit, event**

Counts cycles when no operations are sent in the SME2 unit to the rename from the frontend or from the rename 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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_Stall](#)

### **0x324a CME\_STALL\_FRONTEND, No operation sent for execution due to the SME2 unit frontend, event**

No operation has been issued, because of the frontend. The counter counts on any cycle when no operations are issued while instruction queues are not full.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_retiring](#)
- [cme\\_frontend\\_bound](#)
- [cme\\_frontend\\_cpu\\_bound](#)
- [cme\\_frontend\\_other\\_bound](#)

##### **Metric groups**

[CME\\_Topdown\\_Frontend](#)  
[CME\\_Topdown\\_L1](#)

##### **Functional groups**

[CME\\_Stall](#)

### **0x324b CME\_STALL\_FRONTEND\_CPU, SME2 unit front-end stall due to arbitrated core not sending instructions, event**

The counter counts cycles when CME\_STALL\_FRONTEND increments, due to not enough instructions being sent by the CPU.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_frontend\\_cpu\\_bound](#)

##### **Metric groups**

[CME\\_Topdown\\_Frontend](#)

##### **Functional groups**

[CME\\_Stall](#)

### **0x324c CME\_STALL\_FRONTEND\_OTHER\_CPU, SME2 unit front-end stall due to non-arbitrated cores, event**

The counter counts cycles when CME\_STALL\_FRONTEND increments, due to instructions received by another CPU in the system.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_frontend\\_other\\_bound](#)

##### **Metric groups**

[CME\\_Topdown\\_Frontend](#)

##### **Functional groups**

[CME\\_Stall](#)

### **0x324d CME\_STALL\_BACKEND, No operation sent for execution in the SME2 unit due to the backend, event**

Counts cycles whenever the rename unit in the SME2 unit is unable to send any micro-operations to the backend of the pipeline because of backend resource constraints. Backend resource constraints can include issue stage fullness, execution stage fullness, or other internal pipeline resource fullness.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_retiring](#)
- [cme\\_backend\\_bound](#)
- [cme\\_backend\\_core\\_bound](#)
- [cme\\_backend\\_mem\\_bound](#)
- [cme\\_backend\\_prefetch\\_bound](#)
- [cme\\_iq\\_dp0\\_stall\\_percentage](#)
- [cme\\_iq\\_dp1\\_stall\\_percentage](#)
- [cme\\_iq\\_load\\_stall\\_percentage](#)

##### **Metric groups**

[CME\\_IQ\\_Effectiveness](#)

[CME\\_Topdown\\_Backend](#)

[CME\\_Topdown\\_L1](#)

##### **Functional groups**

[CME\\_Stall](#)

### **0x324e CME\_STALL\_BACKEND\_CORE, SME2 unit backend stall due to the execution units, event**

Counts cycles when CME\_STALL\_BACKEND is set, due to DP issue queues not accepting instructions.

## Related telemetry artifacts

### Metrics

- [cme\\_backend\\_core\\_bound](#)

### Metric groups

[CME\\_Topdown\\_Backend](#)

### Functional groups

[CME\\_Stall](#)

## 0x324f CME\_STALL\_BACKEND\_MEM, SME2 unit backend stall due to the memory system, event

Counts cycles when CME\_STALL\_BACKEND is set, due to Load-Store issue queues not accepting instructions.

## Related telemetry artifacts

### Metrics

- [cme\\_backend\\_mem\\_bound](#)
- [cme\\_backend\\_mem\\_cache\\_bound](#)
- [cme\\_backend\\_mem\\_store\\_bound](#)

### Metric groups

[CME\\_Topdown\\_Backend](#)

### Functional groups

[CME\\_Stall](#)

## 0x3250 CME\_STALL\_BACKEND\_PF, SME2 unit backend stall due to the prefetcher, event

Counts cycles when CME\_STALL\_BACKEND is set, due to prefetcher issue queues not accepting instructions.

## Related telemetry artifacts

### Metrics

- [cme\\_backend\\_prefetch\\_bound](#)

### Metric groups

[CME\\_Topdown\\_Backend](#)

### Functional groups

[CME\\_Stall](#)

## 0x3251 CME\_STALL\_BACKEND\_MEM\_CACHE, SME2 unit memory system stall due to the cache pipelines, event

Counts cycles when CME\_STALL\_BACKEND\_MEM is set, with the oldest instruction in at least one of the load issue queues waiting for cache arbitration.

## Related telemetry artifacts

### Metrics

- [cme\\_backend\\_mem\\_cache\\_bound](#)

### Metric groups

[CME\\_Topdown\\_Backend](#)

### Functional groups

[CME\\_Stall](#)

## 0x3252 CME\_STALL\_BACKEND\_MEM\_STORE, SME2 unit memory system stall due to store backpressure, event

Counts cycles when CME\_STALL\_BACKEND\_MEM is set, with the oldest instruction in at least one of the store issue queues waiting for the merge buffer.

## Related telemetry artifacts

### Metrics

- [cme\\_backend\\_mem\\_store\\_bound](#)

### Metric groups

[CME\\_Topdown\\_Backend](#)

### Functional groups

[CME\\_Stall](#)

## 0x32a8 SSVE\_CONTEXT\_SWITCH, Context switch stall cycles, event

Cycles inside the SME2 unit lost for context switch (new instructions are stalled in decode stage).

Counted for a core by the time arbitration acknowledge is sent to the time first instruction can be executed.

## Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Stall](#)

## 0x32a9 SSVE\_FAST\_CONTEXT\_SWITCH, Stall cycles due to another Core accessing its context, event

Cycles inside the SME2 unit lost because a core loads or stores its context (without causing a real context switch).

## Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_Stall](#)

## 6.9 CME\_General (SME2 GENERAL) events for C1-SME2

General SME2 unit related events.

Summary of events in CME\_General:

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

**Table 6-9: CME\_General events summary**

Code	Mnemonic	Name	Description
0x3246	<a href="#">CME_CYCLES</a>	SME2 unit clock cycles	Counts SME2 unit clock cycles (not timer cycles). The clock measured by this event is defined as...
0x32a4	<a href="#">SSVE_SLOW_INSTR</a>	Slow Streaming SVE instructions	Number of instructions which are considered as slow instructions (communication to the CPU,...
0x32a5	<a href="#">SSVE_GPR_UPDATE</a>	Slow Streaming SVE instructions producing a General purpose register	Number of instructions producing a GPR register (not XZR/WZR) from SME2 unit to the CPU
0x32a6	<a href="#">SSVE_PRED_UPDATE</a>	Slow Streaming SVE instructions producing a predicate register	Number of instructions producing a Predicate register from SME2 unit to the CPU
0x32a7	<a href="#">SSVE_FLAG_UPDATE</a>	Slow Streaming SVE instructions producing condition flags	Number of instructions producing a Flag register (CPSR) from SME2 unit to the CPU

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### 0x3246 CME\_CYCLES, SME2 unit clock cycles, event

Counts SME2 unit clock cycles (not timer cycles). The clock measured by this event is defined as the physical clock driving the SME2 unit.

#### Related telemetry artifacts

##### Metrics

- [cme\\_retiring](#)
- [cme\\_frontend\\_bound](#)
- [cme\\_backend\\_bound](#)
- [cme\\_ipc](#)
- [cme\\_alu\\_port\\_utilization](#)
- [cme\\_mac\\_port\\_utilization](#)
- [cme\\_mmdp\\_port\\_utilization](#)
- [cme\\_mmmv\\_port\\_utilization](#)
- [cme\\_perm\\_port\\_utilization](#)

- [cme\\_st\\_port\\_utilization](#)

#### Metric groups

[CME\\_General](#)  
[CME\\_Port\\_Utilization](#)  
[CME\\_Topdown\\_L1](#)

#### Functional groups

[CME\\_General](#)

### 0x32a4 SSVE\_SLOW\_INSTR, Slow Streaming SVE instructions, event

Number of instructions which are considered as slow instructions (communication to the CPU, serializing instruction)

#### Related telemetry artifacts

##### Metrics

- [streaming\\_slow\\_inst\\_percentage](#)

##### Metric groups

[CME\\_Operation\\_Mix](#)

##### Functional groups

[CME\\_General](#)

### 0x32a5 SSVE\_GPR\_UPDATE, Slow Streaming SVE instructions producing a General purpose register, event

Number of instructions producing a GPR register (not XZR/WZR) from SME2 unit to the CPU

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

##### Functional groups

[CME\\_General](#)

### 0x32a6 SSVE\_PRED\_UPDATE, Slow Streaming SVE instructions producing a predicate register, event

Number of instructions producing a Predicate register from SME2 unit to the CPU

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

##### Functional groups

[CME\\_General](#)

### 0x32a7 SSVE\_FLAG\_UPDATE, Slow Streaming SVE instructions producing condition flags, event

Number of instructions producing a Flag register (CPSR) from SME2 unit to the CPU

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

**Functional groups**

[CME\\_General](#)

## 6.10 CME\_SVE (SME2 SVE) events for C1-SME2

SME2 unit SVE related events.

Summary of events in CME\_SVE:

- Total implemented Common events: 16
- Total Implemented Product ImpDef events: 24
- PMU Only events : 40
- ETE Only events : 0

**Table 6-10: CME\_SVE events summary**

Code	Mnemonic	Name	Description
0x3220	<a href="#">SSVE_FP_SPEC</a>	Operation speculatively executed, Streaming SVE floating-point	The counter counts each Speculatively executed floating-point operation due to an SVE...
0x3225	<a href="#">SSVE_FP_HP_SPEC</a>	Operation speculatively executed, Streaming SVE half-precision floating-point	The counter counts each Speculatively executed half-precision floating-point operation due to an...
0x3226	<a href="#">SSVE_FP_BF16_SPEC</a>	Operation speculatively executed, Streaming SVE BFloat16 floating-point	The counter counts each Speculatively executed BFloat16 floating-point operation due to an SVE...
0x3227	<a href="#">SSVE_FP_SP_SPEC</a>	Operation speculatively executed, Streaming SVE single-precision floating-point	The counter counts each Speculatively executed single-precision floating-point operation due to...
0x3228	<a href="#">SSVE_FP_DP_SPEC</a>	Operation speculatively executed, Streaming SVE double-precision floating-point	The counter counts each Speculatively executed double-precision floating-point operation due to...
0x3234	<a href="#">SSVE_PRED_SPEC</a>	Operation speculatively executed, Streaming SVE predicated	Counts operations counted by SVE_PRED_SPEC, but in Streaming mode only. Note: this counts SME...
0x3235	<a href="#">SSVE_PRED_EMPTY_SPEC</a>	Operation speculatively executed, Streaming SVE predicated with no active predicates	Counts operations counted by SVE_PRED_EMPTY_SPEC, but in Streaming mode only.
0x3236	<a href="#">SSVE_PRED_FULL_SPEC</a>	Operation speculatively executed, Streaming SVE predicated with all active predicates	Counts speculatively executed predicated SVE operations with all predicate elements active,...
0x3237	<a href="#">SSVE_PRED_NOT_FULL_SPEC</a>	Operation speculatively executed, Streaming SVE predicated with no active or partially active predicates	Counts speculatively executed predicated SVE operations with at least one non active predicate...

Code	Mnemonic	Name	Description
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...
0x3239	SSVE_FP_SCALE_OPS_SPEC	Scalable floating-point element ALU operations speculatively executed in Streaming SVE	The counter counts each Streaming SVE floating-point ALU operation counted by SSVE_FP_SPEC that...
0x323a	SSVE_FP_HP_SCALE_OPS_SPEC	Scalable half-precision floating-point element ALU operations in Streaming mode	The counter counts each Streaming SVE floating-point half-precision ALU operation counted...
0x323b	SSVE_FP_BF16_SCALE_OPS_SPEC	Scalable BFloat16 floating-point element ALU operations in Streaming SVE	The counter counts each Streaming SVE floating-point BFloat16 ALU operation counted...
0x323c	SSVE_FP_SP_SCALE_OPS_SPEC	Scalable single-precision floating-point element ALU operations in Streaming SVE	The counter counts each Streaming SVE floating-point single-precision ALU operation counted...
0x323d	SSVE_FP_DP_SCALE_OPS_SPEC	Scalable double-precision floating-point element ALU operations in Streaming SVE	The counter counts each Streaming SVE floating-point double-precision ALU operation counted...
0x323e	SSVE_INT_SCALE_OPS_SPEC	Scalable integer element ALU operations Speculatively executed in Streaming SVE	The counter counts each Streaming SVE integer ALU operation counted by SSVE_INT_SPEC that was...
0x323f	SSVE_LDST_SCALE_OPS_SPEC	Scalable load or store element Operations speculatively executed in Streaming SVE	The counter counts each speculatively executed Memory-read operation or Memory-write operation...
0x3240	SSVE_LD_SCALE_OPS_SPEC	Scalable load element Operations speculatively executed in Streaming SVE	The counter counts each speculatively executed Memory-read operation due to either: <ul style="list-style-type: none"> <li>An SVE...</li> </ul>
0x3241	SSVE_ST_SCALE_OPS_SPEC	Scalable store element Operations speculatively executed in Streaming SVE	The counter counts each speculatively executed Memory-store operation due to either: <ul style="list-style-type: none"> <li>An SVE...</li> </ul>
0x3242	SSVE_LDST_SCALE_BYTES_SPEC	Scalable load and store bytes Speculatively executed in Streaming SVE	The counter counts each speculatively executed Memory-read operation or Memory-write operation...
0x3243	SSVE_LD_SCALE_BYTES_SPEC	Scalable load bytes Speculatively executed in Streaming SVE	The counter counts each speculatively executed Memory-read operation due to either: <ul style="list-style-type: none"> <li>An SVE...</li> </ul>
0x3244	SSVE_ST_SCALE_BYTES_SPEC	Scalable store bytes Speculatively executed in Streaming SVE	The counter counts each speculatively executed Memory-write operation due to either: <ul style="list-style-type: none"> <li>An SVE...</li> </ul>
0x3245	SSVE_LDST_FIXED_BYTES_SPEC	Non-scalable load and store bytes Speculatively executed in Streaming SVE	The counter counts each byte speculatively read or written in SVE streaming due to any of: <ul style="list-style-type: none"> <li>Any...</li> </ul>

Code	Mnemonic	Name	Description
0x32ab	SSVE_UNPRED_SPEC	Operation speculatively executed, Streaming SVE unpredicated	The counter counts each Speculatively executed data-processing, load, or store operation due to...
0x8074	SVE_PRED_SPEC	Operation speculatively executed, SVE predicated	The counter counts each Speculatively executed SIMD data-processing, load, or store operation...
0x8075	SVE_PRED_EMPTY_SPEC	Operation speculatively executed, SVE predicated with no active predicates	The counter counts each Speculatively executed predicated SIMD data-processing, load, or...
0x8076	SVE_PRED_FULL_SPEC	Operation speculatively executed, SVE predicated with all active predicates	The counter counts each Speculatively executed predicated SIMD data-processing, load, or...
0x8077	SVE_PRED_PARTIAL_SPEC	Operation speculatively executed, SVE predicated with partially active predicates	The counter counts each Speculatively executed predicated SIMD data-processing, load, or...
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	Operation speculatively executed, SVE predicated with no active or partially active predicates	Counts speculatively executed predicated SVE operations with at least one non active predicate...
0x80d0	FP_SCALE2_OPS_SPEC	Scalable floating-point matrix element arithmetic operations speculatively executed	The counter counts each speculatively executed scalable floating-point matrix arithmetic...
0x80d2	FP_HP_SCALE2_OPS_SPEC	Scalable floating-point half-precision matrix element arithmetic operations speculatively executed	The counter counts each Streaming SVE half-precision floating-point matrix ALU operation counted...
0x80d3	FP_BF16_SCALE2_OPS_SPEC	Scalable floating-point BFloat16 matrix element arithmetic operations speculatively executed	The counter counts each Streaming SVE BFloat16 floating-point matrix ALU operation counted...
0x80d4	FP_SP_SCALE2_OPS_SPEC	Scalable floating-point single-precision matrix element arithmetic operations speculatively executed	The counter counts each Streaming SVE single-precision floating-point matrix ALU operation...
0x80d6	FP_DP_SCALE2_OPS_SPEC	Scalable floating-point double-precision matrix element arithmetic operations speculatively executed	The counter counts each Streaming SVE double-precision floating-point matrix ALU operation...
0x8381	SME_PRED2_NOT_FULL_SPEC	Operation speculatively executed, SME 2D predicated with at least one inactive element	The counter counts each speculatively executed predicated 2D SME operation which targets the ZA...
0x8384	SME_PRED2_SPEC	Operation speculatively executed, SME 2D predicated	The counter counts each speculatively executed 2D operation which targets the ZA array counted...
0x8385	SME_PRED2_EMPTY_SPEC	Operation speculatively executed, SME 2D predicated with no active element	The counter counts each speculatively executed predicated 2D SME operation which targets the ZA...
0x8386	SME_PRED2_FULL_SPEC	Operation speculatively executed, SME 2D predicated with all active elements	The counter counts each speculatively executed predicated 2D SME operation which targets the ZA...

Code	Mnemonic	Name	Description
0x8387	SME_PRED2_PARTIAL_SPEC	Operation speculatively executed, SME 2D predicated with partially active elements	The counter counts each speculatively executed predicated 2D SME operation which targets the ZA...

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### 0x3220 SSVE\_FP\_SPEC, Operation speculatively executed, Streaming SVE floating-point, event

The counter counts each Speculatively executed floating-point operation due to an SVE instruction, specifically in streaming mode.

#### Related telemetry artifacts

##### Metrics

- [streaming\\_fp\\_op\\_percentage](#)

##### Metric groups

[CME\\_Operation\\_Mix](#)

##### Functional groups

[CME\\_SVE](#)

### 0x3225 SSVE\_FP\_HP\_SPEC, Operation speculatively executed, Streaming SVE half-precision floating-point, event

The counter counts each Speculatively executed half-precision floating-point operation due to an SVE instruction, specifically in streaming mode.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

##### Functional groups

[CME\\_SVE](#)

### 0x3226 SSVE\_FP\_BF16\_SPEC, Operation speculatively executed, Streaming SVE BFloat16 floating-point, event

The counter counts each Speculatively executed BFloat16 floating-point operation due to an SVE instruction, specifically in streaming mode.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

##### Functional groups

[CME\\_SVE](#)

### **0x3227 SSVE\_FP\_SP\_SPEC, Operation speculatively executed, Streaming SVE single-precision floating-point, event**

The counter counts each Speculatively executed single-precision floating-point operation due to an SVE instruction, specifically in streaming mode.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x3228 SSVE\_FP\_DP\_SPEC, Operation speculatively executed, Streaming SVE double-precision floating-point, event**

The counter counts each Speculatively executed double-precision floating-point operation due to an SVE instruction, specifically in streaming mode.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x3237 SSVE\_PRED\_NOT\_FULL\_SPEC, Operation speculatively executed, Streaming SVE predicated 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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x3239 SSVE\_FP\_SCALE\_OPS\_SPEC, Scalable floating-point element ALU operations speculatively executed in Streaming SVE, event**

The counter counts each Streaming SVE floating-point ALU operation counted by SSVE\_FP\_SPEC that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size of the result.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x323a SSVE\_FP\_HP\_SCALE\_OPS\_SPEC, Scalable half-precision floating-point element ALU operations in Streaming mode, event**

The counter counts each Streaming SVE floating-point half-precision ALU operation counted by `ssve_fp_spec` that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size of the result.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x323b SSVE\_FP\_BF16\_SCALE\_OPS\_SPEC, Scalable BFloat16 floating-point element ALU operations in Streaming SVE, event**

The counter counts each Streaming SVE floating-point BFloat16 ALU operation counted by `ssve_fp_spec` that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size of the result.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x323c SSVE\_FP\_SP\_SCALE\_OPS\_SPEC, Scalable single-precision floating-point element ALU operations in Streaming SVE, event**

The counter counts each Streaming SVE floating-point single-precision ALU operation counted by `ssve_fp_spec` that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size of the result.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x323d SSVE\_FP\_DP\_SCALE\_OPS\_SPEC, Scalable double-precision floating-point element ALU operations in Streaming SVE, event**

The counter counts each Streaming SVE floating-point double-precision ALU operation counted by `ssve_fp_spec` that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size of the result.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### 0x323e SSVE\_INT\_SCALE\_OPS\_SPEC, Scalable integer element ALU operations Speculatively executed in Streaming SVE, event

The counter counts each Streaming SVE integer ALU operation counted by `ssve_int_spec` that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size of the result.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### 0x323f SSVE\_LDST\_SCALE\_OPS\_SPEC, Scalable load or store element Operations speculatively executed in Streaming SVE, event

The counter counts each speculatively executed Memory-read operation or Memory-write operation due to either:

- An SVE predicated vector load or store instruction other than a replicating LD1R or LD1RQ instruction.
- An SME vector load or store instruction

The counter increments by the number of elements accessed by the instruction per 128/element\_size vector of the result.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### 0x3240 SSVE\_LD\_SCALE\_OPS\_SPEC, Scalable load element Operations speculatively executed in Streaming SVE, event

The counter counts each speculatively executed Memory-read operation due to either:

- An SVE predicated vector load instruction other than a replicating LD1R or LD1RQ instruction.
- An SME vector load instruction

The counter increments by the number of elements accessed by the instruction per 128/  
element\_size vector of the result.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### **0x3241 SSVE\_ST\_SCALE\_OPS\_SPEC, Scalable store element Operations speculatively executed in Streaming SVE, event**

The counter counts each speculatively executed Memory-store operation due to either:

- An SVE predicated vector store instruction.
- An SME vector store instruction

The counter increments by the number of elements accessed by the instruction per 128/  
element\_size vector of the result.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### **0x3242 SSVE\_LDST\_SCALE\_BYTES\_SPEC, Scalable load and store bytes Speculatively executed in Streaming SVE, event**

The counter counts each speculatively executed Memory-read operation or Memory-write operation due to either:

- An SVE predicated vector load or store instruction other than a replicating LD1R or LD1RQ instruction.
- An SME vector load or store instruction

For each instruction, the counter is incremented by  $(16 / (\text{CSIZE} / \text{MSIZE}))$ , multiplied by the number of transferred vector registers.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### **0x3243 SSVE\_LD\_SCALE\_BYTES\_SPEC, Scalable load bytes Speculatively executed in Streaming SVE, event**

The counter counts each speculatively executed Memory-read operation due to either:

- An SVE predicated vector load instruction other than a replicating LD1R or LD1RQ instruction.

- An SME vector load instruction

For each instruction, the counter is incremented by  $(16 / (\text{CSIZE} / \text{MSIZE}))$ , multiplied by the number of transferred vector registers.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### 0x3244 SSVE\_ST\_SCALE\_BYTES\_SPEC, Scalable store bytes Speculatively executed in Streaming SVE, event

The counter counts each speculatively executed Memory-write operation due to either:

- An SVE predicated vector store instruction.
- An SME vector store instruction

For each instruction, the counter is incremented by  $(16 / (\text{CSIZE} / \text{MSIZE}))$ , multiplied by the number of transferred vector registers.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### 0x3245 SSVE\_LDST\_FIXED\_BYTES\_SPEC, Non-scalable load and store bytes Speculatively executed in Streaming SVE, event

The counter counts each byte speculatively read or written in SVE streaming due to any of:

- Any Advanced SIMD or SVE non-vector load or store operation.
- An SVE replicating LD1R or LD1RQ instruction.

For each instruction, the counter is incremented by the number of bytes transferred per register multiplied by the number of registers transferred multiplied by the number of transfers made per register. For example, the counter counts bytes as follows:

- SVE and Advanced SIMD LD1R instructions increment the counter by  $(\text{MSIZE} / 8)$ .
- SVELD1RQ instructions increment the counter by 16.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### **0x32ab SSVE\_UNPRED\_SPEC, Operation speculatively executed, Streaming SVE unpredicated, event**

The counter counts each Speculatively executed data-processing, load, or store operation due to an Streaming 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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x8074 SVE\_PRED\_SPEC, Operation speculatively executed, SVE predicated, event**

The counter counts each Speculatively executed SIMD data-processing, load, or store operation due to an SVE instruction with a Governing predicate operand that determines the Active elements.

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 the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x8075 SVE\_PRED\_EMPTY\_SPEC, Operation speculatively executed, SVE predicated with no active predicates, event**

The counter counts each Speculatively executed predicated SIMD data-processing, load, or store operation counted by SVE\_PRED\_SPEC where all elements are Inactive.

That is, all elements in the Governing predicate are FALSE.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_SVE](#)

### **0x8076 SVE\_PRED\_FULL\_SPEC, Operation speculatively executed, SVE predicated with all active predicates, event**

The counter counts each Speculatively executed predicated SIMD data-processing, load, or store operation counted by SVE\_PRED\_SPEC where all elements are Active.

That is, all elements in the Governing predicate are TRUE.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_SVE](#)

### **0x8077 SVE\_PRED\_PARTIAL\_SPEC, Operation speculatively executed, SVE predicated with partially active predicates, event**

The counter counts each Speculatively executed predicated SIMD data-processing, load, or store operation counted by SVE\_PRED\_SPEC that is not counted by either SVE\_PRED\_EMPTY\_SPEC or SVE\_PRED\_FULL\_SPEC.

That is, the elements in the Governing predicate are neither all TRUE nor all FALSE.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_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 the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_SVE](#)

### **0x8079 SVE\_PRED\_NOT\_FULL\_SPEC, Operation speculatively executed, SVE predicated with no active or partially active predicates, event**

Counts speculatively executed predicated SVE operations with at least one non active predicate elements.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_SVE](#)

### **0x80d0 FP\_SCALE2\_OPS\_SPEC, Scalable floating-point matrix element arithmetic operations speculatively executed, event**

The counter counts each speculatively executed scalable floating-point matrix arithmetic operation counted by FP\_SPEC.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size by 128/element\_size tile of the result.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

CME\_SVE

### **0x80d2 FP\_HP\_SCALE2\_OPS\_SPEC, Scalable floating-point half-precision matrix element arithmetic operations speculatively executed, event**

The counter counts each Streaming SVE half-precision floating-point matrix ALU operation counted by SME\_FP\_SPEC that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size by 128/element\_size tile of the result.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

CME\_SVE

### **0x80d3 FP\_BF16\_SCALE2\_OPS\_SPEC, Scalable floating-point BFloat16 matrix element arithmetic operations speculatively executed, event**

The counter counts each Streaming SVE BFloat16 floating-point matrix ALU operation counted by SME\_FP\_SPEC that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size by 128/element\_size tile of the result.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

CME\_SVE

### **0x80d4 FP\_SP\_SCALE2\_OPS\_SPEC, Scalable floating-point single-precision matrix element arithmetic operations speculatively executed, event**

The counter counts each Streaming SVE single-precision floating-point matrix ALU operation counted by SME\_FP\_SPEC that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size by 128/element\_size tile of the result.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### **0x80d6 FP\_DP\_SCALE2\_OPS\_SPEC, Scalable floating-point double-precision matrix element arithmetic operations speculatively executed, event**

The counter counts each Streaming SVE double-precision floating-point matrix ALU operation counted by `SME_FP_SPEC` that was speculatively executed.

The counter increments by the number of numerical operations carried out by the instruction per 128/element\_size by 128/element\_size tile of the result.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### **0x8381 SME\_PRED2\_NOT\_FULL\_SPEC, Operation speculatively executed, SME 2D predicated with at least one inactive element, event**

The counter counts each speculatively executed predicated 2D SME operation which targets the ZA array counted by `SME_PRED2_SPEC` where at least one element is Inactive.

That is, at least one element in the Governing predicates is FALSE.

For outer product instructions which are widening, predication is considered with respect to the input element size.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_SVE](#)

### **0x8384 SME\_PRED2\_SPEC, Operation speculatively executed, SME 2D predicated, event**

The counter counts each speculatively executed 2D operation which targets the ZA array counted by `SVE_PRED_SPEC` due to an SME instruction with a Governing predicate operand that determines the Active elements.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_SVE](#)

### **0x8385 SME\_PRED2\_EMPTY\_SPEC, Operation speculatively executed, SME 2D predicated with no active element, event**

The counter counts each speculatively executed predicated 2D SME operation which targets the ZA array counted by SME\_PRED2\_NOT\_FULL\_SPEC where all elements are Inactive.

That is, all elements in the Governing predicates are FALSE.

For outer product instructions which are widening, predication is considered with respect to the input element size.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_SVE](#)

### **0x8386 SME\_PRED2\_FULL\_SPEC, Operation speculatively executed, SME 2D predicated with all active elements, event**

The counter counts each speculatively executed predicated 2D SME operation which targets the ZA array counted by SME\_PRED2\_SPEC where all elements are Active.

That is, all elements in the Governing predicates are TRUE.

For outer product instructions which are widening, predication is considered with respect to the input element size

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_SVE](#)

### **0x8387 SME\_PRED2\_PARTIAL\_SPEC, Operation speculatively executed, SME 2D predicated with partially active elements, event**

The counter counts each speculatively executed predicated 2D SME operation which targets the ZA array counted by SME\_PRED2\_NOT\_FULL\_SPEC where neither all elements are Active nor all elements are Inactive.

That is, the elements in the Governing predicates are neither all TRUE nor all FALSE.

For outer product instructions which are widening, predication is considered with respect to the input element size.

### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

CME\_SVE

## 6.11 CME\_FP\_Operation (CME\_FP\_Operation) events for C1-SME2

SME2 unit Floating-point related events.

Summary of events in CME\_FP\_Operation:

- Total implemented Common events: 11
- Total Implemented Product ImpDef events: 1
- PMU Only events : 12
- ETE Only events : 0

**Table 6-11: CME\_FP\_Operation events summary**

Code	Mnemonic	Name	Description
0x32b0	SME_FP_OTHER_SPEC	Operation speculatively executed, other SME floating-point	The counter counts each speculatively operation counted by SME_FP_SPEC due to an instruction...
0x8010	FP_SPEC	Floating-point operation speculatively executed, including SIMD	The counter counts each Speculatively executed floating-point operation due to an A64...
0x8014	FP_HP_SPEC	Floating-point operation speculatively executed, half precision	Counts speculatively executed half precision floating point operations.
0x8018	FP_SP_SPEC	Floating-point operation speculatively executed, single precision	Counts speculatively executed single-precision floating point operations.
0x801c	FP_DP_SPEC	Floating-point operation speculatively executed, double precision	Counts speculatively executed double-precision floating point operations.
0x80c0	FP_SCALE_OPS_SPEC	Scalable floating-point element ALU operations speculatively executed	Counts speculatively executed scalable floating point ALU operations.
0x8362	SME_FP_BF16_SPEC	Operation speculatively executed, SME BFloat16 floating-point	The counter counts each speculatively executed BFloat16 floating-point operation counted by...
0x836a	SME_FP_SP_SPEC	Operation speculatively executed, SME single-precision floating-point	The counter counts each speculatively executed single-precision floating-point operation counted...
0x8370	SME_FP_ADDSUB_SPEC	Operation speculatively executed, SME floating-point addition or subtraction	The counter counts each speculatively executed floating-point addition or subtraction operation...

Code	Mnemonic	Name	Description
0x8372	<a href="#">SME_FP_FMA_SPEC</a>	Operation speculatively executed, SME floating-point multiply-add or multiply-subtract	The counter counts each speculatively executed floating-point multi-vector multiply and...
0x8374	<a href="#">SME_FP_DOT_SPEC</a>	Operation speculatively executed, SME floating-point dot product	The counter counts each speculatively executed floating-point dot product operation counted...
0x8376	<a href="#">SME_FP_MOPA_SPEC</a>	Operation speculatively executed, SME floating-point outer product and accumulate, or outer product and subtract	The counter counts each speculatively executed floating-point outer product and accumulate, or...

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### 0x32b0 SME\_FP\_OTHER\_SPEC, Operation speculatively executed, other SME floating-point , event

The counter counts each speculatively operation counted by SME\_FP\_SPEC due to an instruction which reads from or writes to any part of the ZA array, which is not counted by SME\_FP\_ADDSUB\_SPEC, SME\_FP\_DOT\_SPEC, SME\_FP\_FMA\_SPEC or SME\_FP\_MOPA\_SPEC

#### Related telemetry artifacts

##### Metrics

- [za\\_fp\\_other\\_percentage](#)

##### Metric groups

[CME\\_Operation\\_Mix](#)

##### Functional groups

[CME\\_FP\\_Operation](#)

### 0x8010 FP\_SPEC, Floating-point operation speculatively executed, including SIMD, event

The counter counts each Speculatively executed floating-point operation due to an A64 scalar, Advanced SIMD, SVE or SME instruction listed in SVE floating-point instructions.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

##### Functional groups

[CME\\_FP\\_Operation](#)

### 0x8014 FP\_HP\_SPEC, Floating-point operation speculatively executed, half precision, event

Counts speculatively executed half precision floating point operations.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_FP\\_Operation](#)

#### **0x8018 FP\_SP\_SPEC, Floating-point operation speculatively executed, single precision, event**

Counts speculatively executed single-precision floating point operations.

##### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_FP\\_Operation](#)

#### **0x801c FP\_DP\_SPEC, Floating-point operation speculatively executed, double precision, event**

Counts speculatively executed double-precision floating point operations.

##### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_FP\\_Operation](#)

#### **0x80c0 FP\_SCALE\_OPS\_SPEC, Scalable floating-point element ALU operations speculatively executed, event**

Counts speculatively executed scalable floating point ALU operations.

##### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_FP\\_Operation](#)

#### **0x8362 SME\_FP\_BF16\_SPEC, Operation speculatively executed, SME BFloat16 floating-point, event**

The counter counts each speculatively executed BFloat16 floating-point operation counted by SME\_FP\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

##### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

### Functional groups

[CME\\_FP\\_Operation](#)

### **0x836a SME\_FP\_SP\_SPEC, Operation speculatively executed, SME single-precision floating-point, event**

The counter counts each speculatively executed single-precision floating-point operation counted by SME\_FP\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

#### **Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### **Functional groups**

[CME\\_FP\\_Operation](#)

### **0x8370 SME\_FP\_ADDSUB\_SPEC, Operation speculatively executed, SME floating-point addition or subtraction, event**

The counter counts each speculatively executed floating-point addition or subtraction operation counted by SME\_FP\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

#### **Related telemetry artifacts**

##### **Metrics**

- [za\\_fp\\_addsub\\_percentage](#)

##### **Metric groups**

[CME\\_Operation\\_Mix](#)

##### **Functional groups**

[CME\\_FP\\_Operation](#)

### **0x8372 SME\_FP\_FMA\_SPEC, Operation speculatively executed, SME floating-point multiply-add or multiply-subtract, event**

The counter counts each speculatively executed floating-point multi-vector multiply and accumulate, or multi-vector multiply and subtract instruction counted by SME\_FP\_SPEC due to an operation which reads from or writes to any part of the ZA array.

#### **Related telemetry artifacts**

##### **Metrics**

- [za\\_fp\\_fma\\_percentage](#)

##### **Metric groups**

[CME\\_Operation\\_Mix](#)

##### **Functional groups**

[CME\\_FP\\_Operation](#)

### **0x8374 SME\_FP\_DOT\_SPEC, Operation speculatively executed, SME floating-point dot product, event**

The counter counts each speculatively executed floating-point dot product operation counted by SME\_FP\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

Related telemetry artifacts

Metrics

- [za\\_fp\\_dot\\_percentage](#)

Metric groups

[CME\\_Operation\\_Mix](#)

Functional groups

[CME\\_FP\\_Operation](#)

**0x8376 SME\_FP\_MOPA\_SPEC, Operation speculatively executed, SME floating-point outer product and accumulate, or outer product and subtract, event**

The counter counts each speculatively executed floating-point outer product and accumulate, or outer product and subtract operations counted by SME\_FP\_SPEC due to an instruction which reads from or writes to any part of the ZA array.

Related telemetry artifacts

Metrics

- [za\\_fp\\_mopa\\_percentage](#)

Metric groups

[CME\\_Operation\\_Mix](#)

Functional groups

[CME\\_FP\\_Operation](#)

## 6.12 CME\_IQ\_Efficiency (SME2 ISSUE QUEUE) events for C1-SME2

SME2 unit Issue Queue Effectiveness related events.

Summary of events in CME\_IQ\_Efficiency:

- Total implemented Common events: 0
- Total Implemented Product ImpDef events: 3
- PMU Only events : 3
- ETE Only events : 0

**Table 6-12: CME\_IQ\_Efficiency events summary**

Code	Mnemonic	Name	Description
0x325d	<a href="#">CME_DISPATCH_STALL_IQ_DP0</a>	SME2 unit dispatch stalled due to IQ full, DP0	Count cycles counted by CME_STALL_BACKEND when at least one operation should be sent to Issue...
0x325e	<a href="#">CME_DISPATCH_STALL_IQ_DP1</a>	SME2 unit dispatch stalled due to IQ full, DP1	Count cycles counted by CME_STALL_BACKEND when at least one operation should be sent to Issue...

Code	Mnemonic	Name	Description
0x325f	CME_DISPATCH_STALL_IQ_LD	SME2 unit dispatch stalled due to IQ full, Load queue	Count cycles counted by CME_STALL_BACKEND when at least one operation should be sent to Issue...

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

#### 0x325d CME\_DISPATCH\_STALL\_IQ\_DP0, SME2 unit dispatch stalled due to IQ full, DP0, event

Count cycles counted by CME\_STALL\_BACKEND when at least one operation should be sent to Issue Queue DP0, and this issue queue is full.

##### Related telemetry artifacts

###### Metrics

- [cme\\_iq\\_dp0\\_stall\\_percentage](#)

###### Metric groups

[CME\\_IQ\\_Effectiveness](#)

###### Functional groups

[CME\\_IQ\\_Efficiency](#)

#### 0x325e CME\_DISPATCH\_STALL\_IQ\_DP1, SME2 unit dispatch stalled due to IQ full, DP1, event

Count cycles counted by CME\_STALL\_BACKEND when at least one operation should be sent to Issue Queue DP1, and this issue queue is full.

##### Related telemetry artifacts

###### Metrics

- [cme\\_iq\\_dp1\\_stall\\_percentage](#)

###### Metric groups

[CME\\_IQ\\_Effectiveness](#)

###### Functional groups

[CME\\_IQ\\_Efficiency](#)

#### 0x325f CME\_DISPATCH\_STALL\_IQ\_LD, SME2 unit dispatch stalled due to IQ full, Load queue, event

Count cycles counted by CME\_STALL\_BACKEND when at least one operation should be sent to Issue Queue LD, and this issue queue is full.

##### Related telemetry artifacts

###### Metrics

- [cme\\_iq\\_load\\_stall\\_percentage](#)

###### Metric groups

[CME\\_IQ\\_Effectiveness](#)

## Functional groups

[CME\\_IQ\\_Efficiency](#)

# 6.13 CME\_Port\_Utilization (SME2 PORT UTILIZATION) events for C1-SME2

Execution unit utilization related events.

Summary of events in CME\_Port\_Utilization:

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

**Table 6-13: CME\_Port\_Utilization events summary**

Code	Mnemonic	Name	Description
0x3260	<a href="#">CME_OP_ALU_ISSUE</a>	SME2 unit ALU operation issued	Count number of operations issued to an ALU execution unit at each cycle.
0x3261	<a href="#">CME_OP_MAC_ISSUE</a>	SME2 unit Multiply-accumulate operation issued	Count number of operations issued to a MAC execution unit at each cycle.
0x3262	<a href="#">CME_OP_PERM_ISSUE</a>	SME2 unit Permutation operation issued	Count number of operations issued to a Permute execution unit at each cycle.
0x3263	<a href="#">CME_OP_ST_ISSUE</a>	SME2 unit Store operation issued	Count number of operations issued to a Store execution unit at each cycle.
0x3264	<a href="#">CME_OP_MMDP_ISSUE</a>	SME2 unit Matrix Multiplication Data-processing operation issued	Count number of operations issued to a Matmul Datapath execution unit at each cycle.
0x3265	<a href="#">CME_OP_MMMV_ISSUE</a>	SME2 unit Matrix Multiplication Move operation issued	Count number of operations issued to a Matmul Move execution unit at each cycle.
0x32aa	<a href="#">CME_OP_DIVSQRT_ISSUE</a>	SME2 unit Divide or Square Root operation issued	Count number of operations issued to a DIV/SQRT execution unit at each cycle.

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

## 0x3260 CME\_OP\_ALU\_ISSUE, SME2 unit ALU operation issued, event

Count number of operations issued to an ALU execution unit at each cycle.

### Related telemetry artifacts

#### Metrics

- [cme\\_alu\\_port\\_utilization](#)

#### Metric groups

[CME\\_Port\\_Utilization](#)

## Functional groups

[CME\\_Port\\_Utilization](#)

### **0x3261 CME\_OP\_MAC\_ISSUE, SME2 unit Multiply-accumulate operation issued, event**

Count number of operations issued to a MAC execution unit at each cycle.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_mac\\_port\\_utilization](#)

##### **Metric groups**

[CME\\_Port\\_Utilization](#)

##### **Functional groups**

[CME\\_Port\\_Utilization](#)

### **0x3262 CME\_OP\_PERM\_ISSUE, SME2 unit Permutation operation issued, event**

Count number of operations issued to a Permute execution unit at each cycle.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_perm\\_port\\_utilization](#)

##### **Metric groups**

[CME\\_Port\\_Utilization](#)

##### **Functional groups**

[CME\\_Port\\_Utilization](#)

### **0x3263 CME\_OP\_ST\_ISSUE, SME2 unit Store operation issued, event**

Count number of operations issued to a Store execution unit at each cycle.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_st\\_port\\_utilization](#)

##### **Metric groups**

[CME\\_Port\\_Utilization](#)

##### **Functional groups**

[CME\\_Port\\_Utilization](#)

### **0x3264 CME\_OP\_MMDP\_ISSUE, SME2 unit Matrix Multiplication Data-processing operation issued, event**

Count number of operations issued to a Matmul Datapath execution unit at each cycle.

#### **Related telemetry artifacts**

##### **Metrics**

- [cme\\_mmdp\\_port\\_utilization](#)

**Metric groups**  
[CME\\_Port\\_Utilization](#)

**Functional groups**  
[CME\\_Port\\_Utilization](#)

**0x3265 CME\_OP\_MMMV\_ISSUE, SME2 unit Matrix Multiplication Move operation issued, event**

Count number of operations issued to a Matmul Move execution unit at each cycle.

**Related telemetry artifacts**

**Metrics**  

- [cme\\_mmmv\\_port\\_utilization](#)

**Metric groups**  
[CME\\_Port\\_Utilization](#)

**Functional groups**  
[CME\\_Port\\_Utilization](#)

**0x32aa CME\_OP\_DIVSQRT\_ISSUE, SME2 unit Divide or Square Root operation issued, event**

Count number of operations issued to a DIV/SQRT execution unit at each cycle.

**Related telemetry artifacts**

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

**Functional groups**  
[CME\\_Port\\_Utilization](#)

## 6.14 CME\_Coherency (SME2 COHERENCY) events for C1-SME2

Coherency related events.

Summary of events in CME\_Coherency:

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

**Table 6-14: CME\_Coherency events summary**

Code	Mnemonic	Name	Description
0x326e	<a href="#">CME_DSNP_HIT</a>	SME2 unit Snoop hit	This event counts each data snoop that hits in a cache outside of the SME2 unit cache.

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### 0x326e CME\_DSNP\_HIT, SME2 unit Snoop hit, event

This event counts each data snoop that hits in a cache outside of the SME2 unit cache.

#### Related telemetry artifacts

There are no related metrics or metric groups because this event is not used in the C1-SME2 Telemetry Specification.

#### Functional groups

[CME\\_Coherency](#)

## 6.15 CME\_System (SME2 SYSTEM) events for C1-SME2

SME2 unit SYSTEM related events.

Summary of events in CME\_System:

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

**Table 6-15: CME\_System events summary**

Code	Mnemonic	Name	Description
0x32ac	<a href="#">CME_DRAM_ACCESS</a>	Access to DRAM due to the SME2 unit	Counts access where the data was sourced from the DRAM.

For a complete list of the events in C1-SME2, see [PMU events cheat sheet for C1-SME2](#) and [PMU events lookup table for C1-SME2](#).

### 0x32ac CME\_DRAM\_ACCESS, Access to DRAM due to the SME2 unit, event

Counts access where the data was sourced from the DRAM.

#### Related telemetry artifacts

##### Metrics

- [cme\\_system\\_dram\\_mem\\_hit\\_ratio](#)

##### Metric groups

[CME\\_System\\_Memory\\_Effectiveness](#)

##### Functional groups

[CME\\_System](#)

# Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant

export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

# Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

## Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

### Product completeness status

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

## Revision history

These sections can help you understand how the document has changed over time.

### Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

#### Document history

Issue	Date	Confidentiality	Change
0400-04	10 September 2025	Non-Confidential	Fourth issue for all revsions of Arm® C1-SME2.
0300-03	30 April 2025	Confidential	Third issue for all revsions of Arm® C1-SME2.
0200-02	30 August 2024	Confidential	Second issue for all revsions of Arm® C1-SME2.
0100-01	1 August 2024	Confidential	First issue for all revsions of Arm® C1-SME2.

### 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 158.

Table 2: Issue 0100-01

Change	Location
First issue for all revisions of Arm® C1-SME2.	-

**Table 3: Differences between Issue 0100-01 and 0200-02**

Change	Location
Second issue for all revisions of Arm® C1-SME2.	-

**Table 4: Differences between Issue 0200-02 and 0300-03**

Change	Location
Third issue for all revisions of Arm® C1-SME2.	-

**Table 5: Differences between Issue 0300-03 and 0400-04**

Change	Location
Fourth issue for all revisions of Arm® C1-SME2.	-
Previous release issue numbering and change information realigned for consistency corresponding release date information remains unchanged	-
Updated product name to C1-SME2	Throughout document
Added CME_Prefetcher_Effectiveness metrics for C1-SME2	<a href="#">CME_Prefetcher_Effectiveness</a>

## Conventions

The following subsections describe conventions used in Arm documents.

### Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: [developer.arm.com/glossary](https://developer.arm.com/glossary).

### Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
<i>italic</i>	Citations.
<b>bold</b>	Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments.  For example:  <pre>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</pre>

Convention	Use
SMALL CAPITALS	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, <b>IMPLEMENTATION DEFINED</b> , <b>IMPLEMENTATION SPECIFIC</b> , <b>UNKNOWN</b> , and <b>UNPREDICTABLE</b> .



We recommend the following. If you do not follow these recommendations your system might not work.



Your system requires the following. If you do not follow these requirements your system will not work.



You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.



This information is important and needs your attention.



This information might help you perform a task in an easier, better, or faster way.



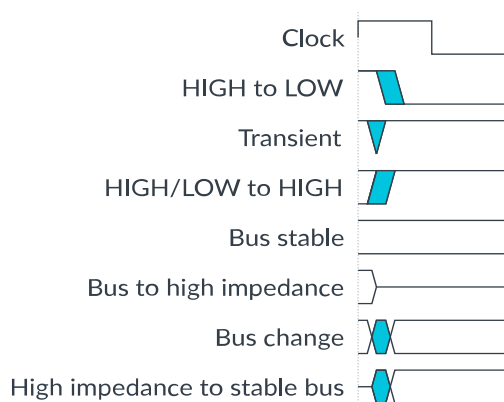
This information reminds you of something important relating to the current content.

### Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

**Figure 1: Key to timing diagram conventions**



## Signals

The signal conventions are:

### Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

### Lowercase n

At the start or end of a signal name, n denotes an active-LOW signal.

# Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Arm documents are available on [developer.arm.com/documentation](https://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
<a href="#">Arm® C1-Scalable Matrix Extension 2 Technical Reference Manual</a>	107831	Non-Confidential
<a href="#">Arm® CPU Telemetry Solution Topdown Methodology Specification</a>	109542	Non-Confidential
<a href="#">Arm® Telemetry Solution GitLab repository</a>	–	Non-Confidential
<a href="#">Arm® Telemetry on Arm Developer</a>	–	Non-Confidential